Kementerian Pendidikan Dan Kebudayaan
Republik Indonesia
2013

# BASIS DATA

Untuk SMK/MAK Kelas XI

Semester 1

## HALAMAN FRANCIS

**Penulis** : **ABDUL MUNIF**
**Editor Materi** : **FARID**
**Editor Bahasa** :
**Ilustrasi Sampul** :
**Desain & Ilustrasi Buku** : **PPPPTK BOE MALANG**



Pusat Pengembangan Pemberdayaan Pendidik dan Tenaga Kependidikan Bidang Otomotif dan Elektronika:

Jl. Teluk Mandar, Arjosari Tromol Pos 5, Malang 65102, Telp. (0341) 491239, (0341) 495849, Fax. (0341) 491342, Surel: vedcmalang@vedcmalang.or.id, Laman: www.vedcmalang.com

**DISKLAIMER (*DISCLAIMER*)**

Penerbit tidak menjamin kebenaran dan keakuratan isi/informasi yang tertulis di dalam buku tek ini. Kebenaran dan keakuratan isi/informasi merupakan tanggung jawab dan wewenang dari penulis.

Penerbit tidak bertanggung jawab dan tidak melayani terhadap semua komentar apapun yang ada didalam buku teks ini. Setiap komentar yang tercantum untuk tujuan perbaikan isi adalah tanggung jawab dari masing-masing penulis.

Setiap kutipan yang ada di dalam buku teks akan dicantumkan sumbernya dan penerbit tidak bertanggung jawab terhadap isi dari kutipan tersebut. Kebenaran keakuratan isi kutipan tetap menjadi tanggung jawab dan hak diberikan pada penulis dan pemilik asli. Penulis bertanggung jawab penuh terhadap setiap perawatan (perbaikan) dalam menyusun informasi dan bahan dalam buku teks ini.

Penerbit tidak bertanggung jawab atas kerugian, kerusakan atau ketidaknyamanan yang disebabkan sebagai akibat dari ketidakjelasan, ketidaktepatan atau kesalahan didalam menyusun makna kalimat didalam buku teks ini.

Kewenangan Penerbit hanya sebatas memindahkan atau menerbitkan mempublikasi, mencetak, memegang dan memproses data sesuai dengan undang-undang yang berkaitan dengan perlindungan data.

Katalog Dalam Terbitan (KDT)
Basis Data, Edisi Pertama 2013
Kementerian Pendidikan & Kebudayaan
Direktorat Jenderal Peningkatan Mutu Pendidik & Tenaga Kependidikan, th. 2013: Jakarta

# KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan yang Maha Esa atas tersusunnya buku teks ini, dengan harapan dapat digunakan sebagai buku teks untuk siswa Sekolah Menengah Kejuruan (SMK) Bidang Studi Teknik Elektronika.

Penerapan kurikulum 2013 mengacu pada paradigma belajar kurikulum abad 21 menyebabkan terjadinya perubahan, yakni dari pengajaran (*teaching*) menjadi BELAJAR (*learning*), dari pembelajaran yang berpusat kepada guru (*teachers-centered*) menjadi pembelajaran yang berpusat kepada peserta didik (*student-centered*), dari pembelajaran pasif (*pasive learning*) ke cara belajar peserta didik aktif (*active learning-CBSA*) atau *Student Active Learning-SAL*.

Buku teks ″Basis Data″ ini disusun berdasarkan tuntutan paradigma pengajaran dan pembelajaran kurikulum 2013 diselaraskan berdasarkan pendekatan model pembelajaran yang sesuai dengan kebutuhan belajar kurikulum abad 21, yaitu pendekatan model pembelajaran berbasis peningkatan keterampilan proses sains.

Penyajian buku teks untuk Mata Pelajaran ″Basis Data″ ini disusun dengan tujuan agar supaya peserta didik dapat melakukan proses pencarian pengetahuan berkenaan dengan materi pelajaran melalui berbagai aktivitas proses sains sebagaimana dilakukan oleh para ilmuwan dalam melakukan eksperimen ilmiah (penerapan scientifik), dengan demikian peserta didik diarahkan untuk menemukan sendiri berbagai fakta, membangun konsep, dan nilai-nilai baru secara mandiri.

Kementerian Pendidikan dan Kebudayaan, Direktorat Pembinaan Sekolah Menengah Kejuruan, dan Direktorat Jenderal Peningkatan Mutu Pendidik dan Tenaga Kependidikan menyampaikan terima kasih, sekaligus saran kritik demi kesempurnaan buku teks ini dan penghargaan kepada semua pihak yang telah berperan serta dalam membantu terselesaikannya buku teks siswa untuk Mata Pelajaran basis data kelas XI /Semester 1 Sekolah Menengah Kejuruan (SMK).

Jakarta, 12 Desember 2013

Menteri Pendidikan dan Kebudayaan

Prof. Dr. Mohammad Nuh, DEA

DAFTAR ISI

**GLOSARIUM**

**Abstraksi data** adalah merupakan tingkatan atau level bagaimana melihat data dalam sistem basis data, sejumlah konsep yang digunakan untuk membuat diskripsi struktur basis data, diwujudkan dalam pemodelan data, melalui diskripsi tersebutdapat ditentukan jenis data dan hubungannya deangan data lain

***Attribute*** adalah merupakan karakteristik dari entitas atau relationship, yang menyediakan penjelasan detail entitas atau relationship tersebut. Dalam penerapannya (level fisik) atribut merupakan field atau kolom dari sebuah tabel.

***Basis Data***: adalah kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundancy) yang tidak perlu, untuk memenuhi berbagai kebutuhan

**Entitas** adalah obyek yang mewakili sesuatu dalam dunia nyata dan dapat dibedakan antara satu dengan lainnya (unique). Entitas dapat berupa:Data Fisik (seperti mobil, rumah, manusia, pegawai), abstrak atau konsep (seperti department, pekerjaan, mata pelajaran) dan Kejadian(pembelian, penjualan, peminjaman)

**Key attribute**adalah suatu atribut yang menandakan kunci dari suatu entitas dan bersifat atau mempunyai nilai unik sehingga dapat digunakan untuk membedakan data pada suatu baris atau record dengan baris lain pada suatu entitas

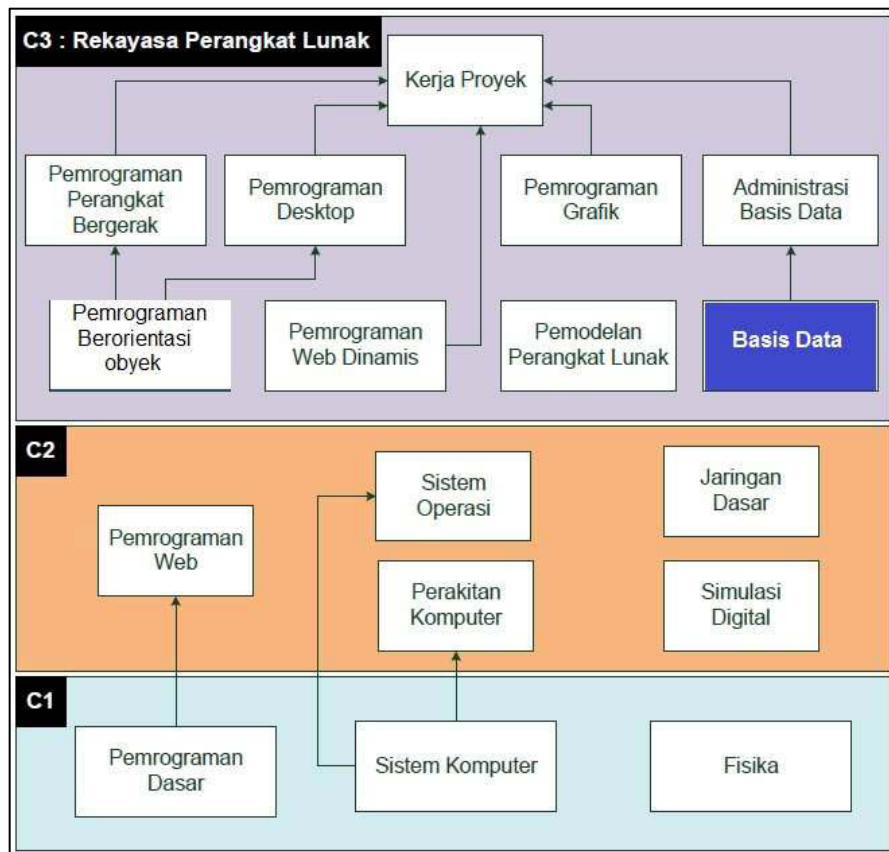
**Pemodelan data**adalah merupakan sarana untuk melakukan abstraksi data dan sejumlah konsep untuk membuat diskripsi stuktur basis data. Terdapat sejumlah cara dalam merepresentasikan model dalam perancangan basis data. Secara umum dikelompokkan menjadi dua yaitu :*Object based logical model dan Record-based logical model*

***Sistem manajemen basis data (SMBD)*** adalahatau *data base mangemen system* (DBMS) merupakan sebuah tatanan (keterpaduan) yang terdiri atas sejumlah komponen-komponen fungsional (komputer) yang saling berhubungan secara bersama-sama, bertujuan untuk memenuhi suatu proses atau pekerjaaan tertentu, program aplikasi yang dibuat dan bekerja dalam satu system

**Skema basis data** atau abstraksi data merupakan diskripsi dari basis data yang spesifikasinya ditentukan dalam tahap perancangan. Arsitektur tiga skema basis data meliputi tiga level yaitu: Level Internal atau skema internal, Level Konseptual (skema konseptual) dan Level eksternal (skema eksternal atau view),

***Struktur atau arsitektur basis data*** kumpulan dari komponen-komponen basis data dan hubungan antar komponen tersebut, merupakan serangkaian pengetahuan tentang File, table, field, record indeks, abstraksi dan pemodelan data serta serangkaian konsep yang digunakan untuk membuat diskripsi struktur basis data.

## PETA KEDUDUKAN BAHAN AJAR

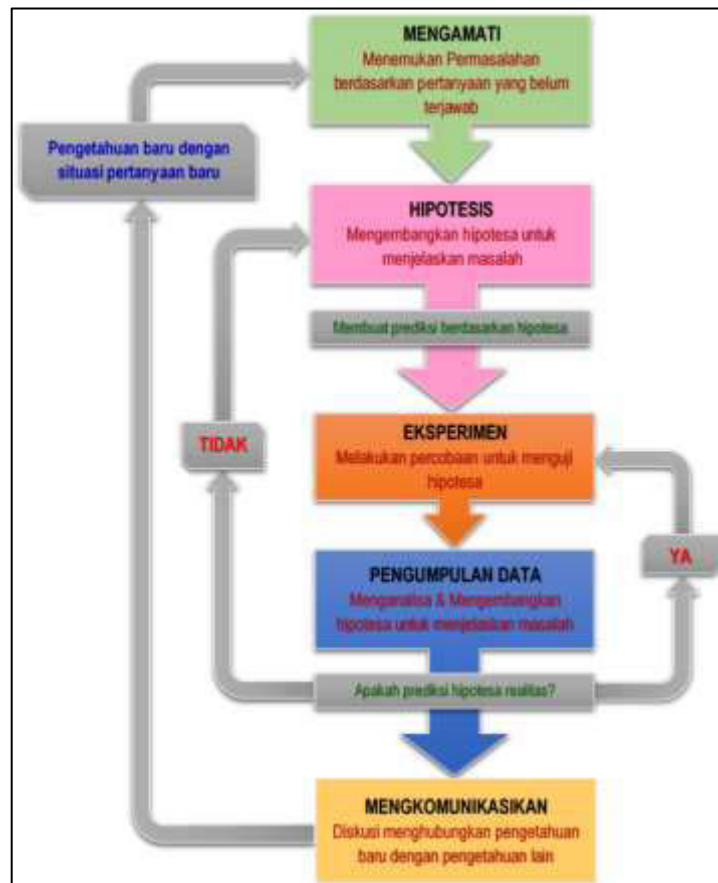
Peta kedudukan bahan ajar merupakan suatu diagram yang menjelaskan struktur mata pelajaran dan keterkaitan antar mata pelajaran dalam satu kelompok bidang studi keakhlian. Gambar 1 menjelaskan peta kedudukan bahan ajar untuk program studi keahlian Rekayasa perangkat lunak. Kelompok C1 merupakan kelompok mata pelajaran wajib dasar bidang studi keahlian. C2 merupakan kelompok mata pelajaran wajib dasar program keahlian dan C3merupakan kelompok mata pelajaran wajib paket keakhlian.

Gambar 1. Peta Kedudukan Bahan Ajar Kelompok C2 Mata Pelajaran Basis Data

# I. BAB I PENDAHULUAN

## A. Diskripsi.

Basis data adalah salah satu mata pelajaran paket Rekayasa perangkat Lunak (RPL) pada program keahlian Teknik Komputer dan Informatika (TKI). Berdasarkan struktur kurikulum mata pelajaran sistem operasi disampaikan di kelas XI semester satu dan semester dua serta kelas XII semester 1, masing-masing 4 jam pelajaran.

Dalam suatu organisasi industri keberadaan data dan informasi memegang peranan yang penting. Data merupakan karakteristik dari suatu obyek-obyek dalam organisasi. Informasi merupakan pengolahan berbagai ragam data yang mempunyai arti tertentu dan sangat bermanfaat untuk kelangsungan hidup organisasi. Dalam pengolahan data dibutuhkan sistem pengelolaan yang melibatkan berbagai macam ragam data dan berasal dari berbagai macam sumber. Pemahaman terhadap basis data dan ketrampilan dalam mengelola sistem basis data sangat dibutuhkan sejalan dengan kebutuhan teknologi informasi dan komunikasi untuk membantu proses atau aktifitas organisasi.

Pembelajaran sistem operasi ini menggunakan metode *pendekatan scientifik.* Dalam pendekatan ini praktikum atau eksperimen berbasis sains merupakan bidang pendekatan ilmiah dengan tujuan dan aturan khusus, dimana tujuan utamanya adalah untuk memberikan bekal ketrampilan yang kuat dengan disertai landasan teori yang realistis mengenai fenomena yang akan kita amati. Ketika suatu permasalahan yang hendak diamati memunculkan pertanyaan-pertanyaan yang tidak bisa terjawab, maka metode eksperimen ilmiah hendaknya dapat memberikan jawaban melalui proses yang logis. Proses-proses dalam pendekatan scientifik meliputi beberapa tahapan (gambar 3) yaitu: mengamati, hipotesis atau menanya, mengasosiasikan atau eksperimen, mengumpulkan atau analisa data dan mengkomunikasikan. Proses belajar pendekatan eksperimen pada hakekatnya merupakan proses berfikir ilmiah untuk membuktikan hipotesis dengan logika berfikir.

Gambar 3. Diagram Proses Metode Scientifik-Eksperimen Ilmiah

**B. Prasyarat.**

Untuk kelancaran pencapaian kompetensi dalam mata pelajaran basis data ini dibutuhkan beberapa peryaratan baik pengetahuan maupun ketrampilan dasar. Persyaratan tersebut antara lain ialah: Peserta didik telah menguasai dasar-dasar pemrograman. Konsep dan implementasi pemrograman ini dibutuhkan untuk mendukung sistem pengelolaan basis data yang akan diimplementasikan store prosedur atau administrasi basis data. Disamping itu peserta didik mempunyai kompetensi dalam hal pemanfaatan teknologi informasi, seperti mengoperasikan hardware komputer dan mengoperasikan perangkat lunak aplikasi. Perangkat lunak aplikasi tersebut antar lain ialah pengolah data untuk menganalisis data hasil eksperimen, pengolah kata untuk membuat laporan dan aplikasi presentasi untuk mengkomunikasikan dan mempresentasikan hasil laporan.

**C. Petunjuk Penggunaan.**

Buku pedoman siswa ini disusun berdasarkan kurikulum 2013 yang mempunyai ciri khas penggunaan metode scientifik. Buku ini terdiri dari dua bab yaitu bab satu pendahuluan dan bab dua pembelajaran. Dalam bab pendahuluan beberapa yang harus dipelajari peserta didik adalah diskripsi mata pelajaran yang berisi informasi umum, rasionalisasi dan penggunaan metode scientifik. Selanjutnya pengetahuan tentang persyaratan, tujuan yang diharapkan, kompetensi inti dan dasar yang akan dicapai serta test kemampuan awal.

Bab dua menuntun peserta didik untuk memahami deskripsi umum tentang topik yang akan dipelajari dan rincian kegiatan belajar sesuai dengan kompetensi dan tujuan yang akan dicapai. Setiap kegiatan belajar terdiri dari tujuan dan uraian materi topik pembelajaran, tugas serta test formatif. Uraian pembelajaran berisi tentang diskripsi pemahaman topik materi untuk memenuhi kompetensi pengetahuan. Uraian pembelajaran juga menjelaskan deskripsi unjuk kerja atau langkah-langkah logis untuk memenuhi kompetensi skill.

Tugas yang harus dikerjakan oleh peserta didik dapat berupa tugas praktek, eksperimen atau pendalaman materi pembelajaran. Setiap tugas yang dilakukan melalui beberapa tahapan scientifik yaitu : 1) melakukan pengamatan setiap tahapan unjuk kerja 2) melakukan praktek sesuai dengan unjuk kerja 3) mengumpulkan data yang dihasilkan setiap tahapan 4) menganalisa hasil data menggunakan analisa diskriptif 5) mengasosiasikan beberapa pengetahuan dalam uraian materi pembelajaran untuk membentuk suatu kesimpulan 6) mengkomunikasikan hasil dengan membuat laporan portofolio. Laporan tersebut merupakan tagihan yang akan dijadikan sebagai salah satu referensi penilaaian.

**D. Tujuan Akhir.**

Setelah mempelajari uraian materi dalam bab pembelajaran dan kegiatan belajar diharapkan peserta didik dapatmemiliki kompetensi sikap, pengetahuan dan ketrampilan yang berkaitan dengan materi:

- ✓ Sistem manajemen basis data
- ✓ Struktur hirarki sistem basis data
- ✓ Entity relationship diagram
- ✓ Teknik Normalisasi data
- ✓ Standar query language

**E. Kompetensi Inti Dan Kompetensi Dasar**

1. Kompetensi Inti 1 : Menghayati dan mengamalkan ajaran agama yang dianutnya.

   Kompetensi Dasar :

   1.1. Memahami nilai-nilai keimanan dengan menyadari hubungan keteraturan dan kompleksitas alam dan jagad raya terhadap kebesaran Tuhan yang menciptakannya

   1.2. Mendeskripsikan kebesaran Tuhan yang menciptakan berbagai sumber energi di alam

   1.3. Mengamalkan nilai-nilai keimanan sesuai dengan ajaran agama dalam kehidupan sehari-hari.

2. Kompetensi Inti 2: Menghayati dan Mengamalkan perilaku jujur, disiplin, tanggung jawab, peduli (gotong royong, kerjasama, toleran, damai), santun, responsif dan proaktif dan menunjukan sikap sebagai bagian dari solusi atas berbagai permasalahan dalam berinteraksi secara efektif dengan lingkungan sosial dan alam serta dalam menempatkan diri sebagai cerminan bangsa dalam menempatkan diri sebagai cerminan bangsa dalam pergaulan dunia.

   Kompetensi Dasar:

   2.1. Menunjukkan perilaku ilmiah (memiliki rasa ingin tahu; objektif; jujur; teliti; cermat; tekun; hati-hati; bertanggung jawab; terbuka; kritis; kreatif; inovatif dan peduli lingkungan) dalam aktivitas sehari-hari sebagai wujud implementasi sikap dalam melakukan percobaan dan berdiskusi

   2.2. Menghargai kerja individu dan kelompok dalam aktivitas sehari-hari sebagai wujud implementasi melaksanakan percobaan dan melaporkan hasil percobaan.

3. Kompetensi Inti 3: Memahami,menerapkan dan menganalisis pengetahuan faktual, konseptual dan prosedural berdasarkan rasa ingintahunya tentang ilmu pengetahuan, teknologi, seni, budaya, dan humaniora dalam wawasan kemanusiaan, kebangsaan, kenegaraan, dan peradaban terkait penyebab fenomena dan kejadian dalam bidang kerja yang spesifik untuk memecahkan masalah.

   Kompetensi Dasar:

   3.1. Memahami struktur hirarki basis data.

   3.2. Memahami bentuk diagram hubungan antar entitas.

3.3. Menganalisisteknik normalisasi basis data.
3.4. Memahami prinsip ketergantungan fungsional dalam perancangan basis data.
3.5. Memahami databasemanagement system (DBMS) sederhana
3.6. Memahami bahasa untuk mengelola basis data.

4. Kompetensi Inti 4: Mengolah, menalar, dan menyaji dalam ranah konkret dan ranah abstrak terkait dengan pengembangan dari yang dipelajarinya di sekolah secara mandiri, dan mampu melaksanakan tugas spesifik dibawah pengawasan langsung.

   Kompetensi Dasar:

   4.1. Menyajikan hasil bentuk struktur hirarki basis data.
   4.2. Menyajikan hasil hubungan keterkaitan antar data dalam diagram ERD.
   4.3. Menyajikan hasil perancangan sistem basis data menggunakan teknik normalisasi data.
   4.4. Menyajikan basis data hasil perancangan menggunakan prinsip-prinsip ketergantungan fungsional.
   4.5. Menyajikan karakteristik beberapa aplikasi DBMS.
   4.6. Menyajikan hasil analisis instruksi pengolahan basis data.

## F. Cek Kemampuan Awal

1. Jelaskan beberapa pengertian atau definisi basis data secara istilah?
2. Jelaskan pengertian Sistem manajemen basis data ?
3. Jelaskan secara singkat definisi struktur atau arsitektur basis data ?
4. Jelaskan secara singkat dan berikan contoh pengertian entitas, atribut dan key atribut ?
5. Jelaskan pengertian tentang tabel, record, colom, indeks, dan batasan partisipasi
6. Jelaskan secara singkat definisi entity relationship diagram (ERD)
7. Jelaskan secara singkat pengertian model struktur hirarki basis data?
8. Jelaskan pengertian model struktur jaringan basis data ?
9. Jelaskan ragam relasi dalam sistem basis data ?
10. Jelaskan secara singkat algoritma mapping ERD ke tabel relasional.

# II. BAB 2 PEMBELAJARAN

## A. Deskripsi

Basis data adalah merupakan kumpulan data yang saling berhubungan yang disimpan secara bersama, sedemikian rupa dan tanpa pengulangan (redundancy) yang tidak perlu, untuk memenuhi berbagai kebutuhan. Ruang lingkup mata pelajaran inimenitik-beratkan pada strategi perancangan dan pembuatan sistem basis data.

Topik materi yang dipelajarai dalam mata pelajaran ini antara lain adalah: struktur hirarki basis data, ketergantungan fungsional, *entity relationship diagram* (ERD), teknik normalisasi data, standar query language (SQL) dan aplikasi sistem manajemen basis data atau database managemen sistem (DBMS)

Topik hirarki basis data menjelaskan tentang konsep basis data, arsitektur basis data, model struktur hirarki basis data dan struktur model jaringan basis data. Konsep basis data mempelajari tentang definisi basis data, tujuan dan manfaat basis data, pengertian sistem manajemen basis data dan operasi dasar dalam manajemen basis data.

Topik ketergantungan fungsional menguraikan materi tentang ragam relasi basis data, batasan partisipasi (constraint) dan dependency. Relasi basis data meliputi relasi one to one, relasi one to many, relasi many to many dan relasi ternary. Batasan partisipasi meliputi partisipasi total dan partisipasi parsial.

Topik *entity relationship diagram*menjelaskan tentang perancangan diskripsi sistem basis data,identifikasi entitas, identifikasi atribute dan relasi, membuat ER diagramserta memetakan ER ke tabel relasional.

Topik Standar query language (SQL) menjelaskan tentang pemakaian bahasa query untuki mengakses data yang meliputi data definition language (DD) dan data manipulation language (DML).

## B. Kegiatan Belajar

Kegiatan belajar menjelaskan tentang aktifitas pembelajaran yang dilakukan peserta didik,meliputi mempelajari uraian materi, mengamati berbagai contoh yang diberikan, mengerjakan test formatif dan tugas atau eksperimen dari proses mengamati sampai menyusun laporan

**Kegiatan belajar 1 : struktur Basis Data (konsep basis data)**

**a. Tujuan Pembelajaran.**

Setelah mengikuti kegiatan belajar 1 ini diharapkan peserta didik dapat:

- ✓ Memahami konsep manajemen basis data.
- ✓ Mengoperasikan contoh aplikasi basis data.

**b. Uraian materi.**

**1) Definisi Basis Data**

Secara umum untuk menjelaskan tentang pengertian basis data dapat ditinjau dari dua sisi,pengertian secara kharfiah dan pengertian secara istilah. Menurut pengertian secara kharfiah, basis data terdiri dari dua kata yaitu basis dan data. Basis dapat diartikan sebagai suatu markas atau gudang, tempat bersarang atau tempat berkumpul.Data dapat diartikan merupakan representasi dari fakta dunia yang mewakili suatu obyek (manusia, barang, peristiwa, keadaan dsb) yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya.Adapun menurut pengertian secara istilah, terdapat beberapa definisi yaitu sebagai berikut :

- ✓ Himpunan kelompok data (arsip) yang saling berhubungan yangdiorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah
- ✓ Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundancy) yang tidak perlu, untuk memenuhi berbagai kebutuhan
- ✓ Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan tertentu.
- ✓ Kumpulan data, yang dapat digambarkan sebagai aktifitas dari satu atau lebih organisasi yang berelasi.

Menurut Elmasri, penggunaan istilah basis data lebih dibatasi pada arti implisit yang khusus mempunyai beberapa pengertian, yaitu :

- ✓ Basis data merupakan penyajian suatu aspek dari dunia nyata (*real word* atau *miniworld*). Misalnya basis data perbankan, perpustakaan, pertanahan, perpajakan

- ✓ Basis data merupakan kumpulan data dari berbagai sumber yang secara logika mempunyai arti implicit. Sehingga apabila data terkumpul secara acak dan tanpa mempunyai arti, tidak dapat disebut basis data.
- ✓ Basis data perlu diancanag, dibangun dan data dikumpulkan untuk suatu tujuan tertentu.
- ✓ Basis data dapat digunakan oleh beberapa pemakai dan beberapa aplikasi yang sesuai dengan kepentingan pemakai.

**2) Komponen Basis data.**

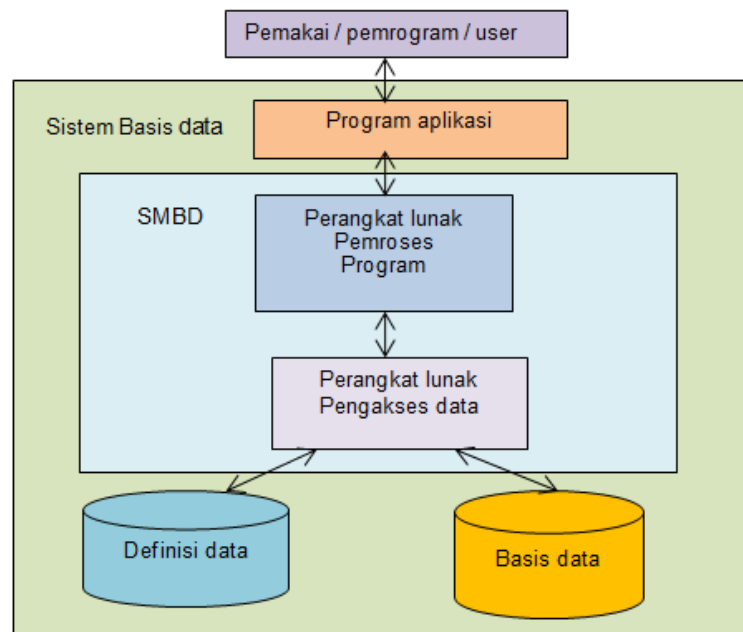
Basis data adalah merupakan suatu sistem yang dibangun oleh beberapa komponen diantaranya ada enam komponen pokok antara lain ialah:

1. Perangkat keras (hardware) dalam sistem komputer. Dalam sistem pengolahan basis data digital perangkat utama sebagai pengolah data dalah komputer.
2. Perangkat Lunak Aplikasi (software) lain yang mendukung dan bersifat opsional. Perangkat lunak digunakan untuk mendukung proses pengelolaan basis data. Misal: bahasa pemrograman C, basic pascal.
3. Sistem Operasi (operating system). Sistem operasi merupakan perangkat lunak yang digunakan untuk mengelola aplikasi basis data dan penggunaan sumberdaya komputer.
4. Basis data data lain yang mempunyai keterkaitan dan hubungan dengan basis data itu sendiri. Berisi atau memiliki objek-objek basis data seperti file, table, indeks . Mempunyai disfinisi struktur baik untuk basis data maupun objek-objek secara detail.
5. Sistem Pengelola Basis Data Database Management System atau database managemen system (DBMS). Merupakan program aplikasi untuk pengelolaan basis data, seperti Microsoft acces, oracle dan lian-lain
6. Pemakai (user), yaitu pengguna yang terlibat dalam pengelolaan basis dan penggunaan basis data.

**3) Sistem manajemen basis Data**

Sistem manajemen basis data adalah merupakan sebuah tatanan (keterpaduan) yang terdiri atas sejumlah komponen-komponen fungsional

(komputer) yang saling berhubungan secara bersama-sama, bertujuan untuk memenuhi suatu proses atau pekerjaaan tertentu. Sistem ini merupakan gabungan antara basis data dan kumpulan program atau perangkat lunak DBMS (d*atabase management system).*

*DBMS a*dalah program aplikasi yang dibuat dan bekerja dalam satu system. *DBMS* didesain untuk membantu dalam hal pemeliharaan dan utilitas kumpulan data dalam jumlah besar. DBMS dapat menjadi alternatif penggunaan secara khusus untuk aplikasi, misalnya penyimpanan data dalam field dan menulis kode aplikasi yang spesifik untuk pengaturannya. Kumpulan file (table) yang saling berhubungan dalam di sebuah komputer dan sekumpulan program yang memungkinkan beberapa pemakai dan atau program lain untuk mengakses dan memanipulasi file-file atau table-tabel tersebut.

Gambar 1. Konsep system basis data dan DBMS

### 4) Tujuan dan Manfaat Penggunaan basis data

Kesuksesan suatu organisasi bergantung pada kemampuannya menangkap data secara akurat dan tepat waktu. Hal tersebut berkaitan dengan operasi dan pengaturan data secara efektif, maupun penggunaan data untuk keperluan analisis untuk kebutuhan pendukung keputusan. Kemampuan untuk mengatur atau mengolah sejumlah data, dan kecepatan untuk mencari informasi yang relevan, adalah aset yang sangat penting bagi suatu organisasi. Untuk

mendapatkan himpunan data yang besar dan kompleks, user harus memiliki alat bantu (*tools*) yang akan menyederhanakan tugas manajemen data dan mengekstrak informasi yang berguna secara tepat waktu.Beberapa tujuan penggunaan basis dataadalah sebagai berikut :

1. Kecepatan dan Kemudahan (*Speed*) , melalui basis data diharapkan pengguna dapat melakukan penyimpanan, perubahan dan menampilkan kembali dengan cepat dan mudah.
2. Efisiensi Ruang Penyimpanan (*Space*). Penggunaan basis data mampu mengurangi pengulangan atau redundansi data. Hal ini dapat dilakukan dengan menerapkan sejumlah pengkodean atau dengan membuat relasi-relasi (dalam bentuk file) antara kelompok data yang saling berhubungan.
3. Keakuratan (*Accuracy*), melalui basis data data keakuratan data lebih terjaga dengan menerapkan aturan dan batasan tertentu (*constraint*), tipe data, domain data dan keunikan data
4. Ketersediaan (*Availability*). Dengan basis data data yang sudah tidak dipakai dapat dipisahkan dari sistem database yang sedang aktif. Hal ini dapat dilakukan dengan cara penghapusan atau memindahkannya ke media backup untuk menghemat ruang penyimpanan. Selain itu dapat memanfaatkan teknologi jaringan komputer agar data yang berada di suatu lokasi atau cabang daat juga diakses oleh lokasi atau cabang lainnya.
5. Kelengkapan (*Completeness*). Agar data yang dikelola senantiasa lengkap baik relatif terhadap kebutuhan pemakai maupun terhadap waktu. Hal ini dapat dilakukan melaluipenambahan record-record data, perubahan struktur basis data, menambah field pada tabel atau menambah tabel baru.
6. Keamanan (*Security*). Walaupun tidak semua sistem basis data menerapkannya, keamanan dalam penggunaan basis data diperlakukan pada sistem yang besar dan serius. Dengan penerapan ini, setiap pengguna dibedakan hak aksesnya; yakni ditentukan obyek-obyek mana saja yang bisa diakses dan proses apa saja yang bisa dia dilakukan.
7. Kebersamaan (*Sharability*). Agar data yang dikelola oleh sistem mendukung lingkungan multiuser (banyak pemakai) dengan menjaga / menghindari munculnya problem baru seperti *inkonsistensi data* (karena terjadi perubahan data yang dilakukan oleh beberapa user dalam waktu

yang bersamaan) atau kondisi *deadlock* (karena ada banyak pemakai yang saling menunggu untuk menggunakan data).

**5) Pengguna dalam Basis data**

Pada tingkat pemakai, data base dikelompokkan menjadi beberapa tingkat pemakai yaitu antara lain sebagai berikut:

1. Database Administrator, ialahmanusia yang mengorganisasi seluruh sistem basis data. Database adaministrator imemiliki tanggung jawab penuh dalam manajemen database meliputi: pengaturan hak akses, koordinasi dan monitoring serta bertanggung jawab terhadap kebutuhan hardware dan software. Dalam pekerjaannya biasanya dibantu oleh staf Admin.
2. *Database Designer*, adalah manusia yang bertugas merancang dan mengembangkan database. Database designer bertanggung jawab dalam identifikasi data yang tersimpan dalam database, menentukan struktur data yang tepat untuk disimpan dalam database. Database designer memerlukan koordinasi akan kebutuhan user database.
3. *Application Programmer,* ialah penggunaa yang berinteraksi dengan basis data melalui *Data Manipulation Language* (DML). DML meliputi program yang ditulis dalam bahasa pemrograman induk yang dipakai.
4. *End user*, adalah adalah pengguna yang memanfaatkan atau membutuhkan akses ke database melalui query, manambah, merubah menghapus maupun membuat *report database*. *End user*dapat dikategorikan:
   a) *Casual end users* atau pengguna tak tetap atau user mahir. Pengguna yang tidak selalu mengakses database, tapi kadang memerlukan informasi terbaru. Berinteraksi dengan sistem tanpa modul program, hanya menggunakan *query* (untuk akses dan manipulasi data) yang telah disediakan oleh DBMS.
   b) *Natïve* atau *parametric end users* atau user umum. Pengguna yang pekerjaan selalu konstan yaitu melakukan query dan update data. Misalnya:*bank teller*, pegawai reservasi. Pengguna ini berinteraksi dg sistem melalui pemanggilan suatu program aplikasi permanen (executable) yang telah dibuat sebelumnya oleh programmer.

c) User Khusus (Specialized User). Pengguna yang menulis aplikasi basis data *non konvensional* untuk keperluan khusus yang bisa saja mengakses basis data dengan atau tanpa DBMS yang bersangkutan.

d) *Sophisticated end users*. pengguna yang melengkapi kebutuhan database user, seperti engineer, scientist, business analyst.

e) Stand-alone users. penggunauser yang mengelola personal database.

5. *System Analyst*, ialah pengguna yang merencanakan dan menentukan kebutuhan sistem.
6. *Application Programmers* (Software Engineering), ialah pengguna tanggungjawabnya berhubungan dengan kebutuhan koneksi database.
7. *Worker behind the scene, ialah pengguna* yang tidak tertarik pada database, tetapi lebih cenderung pada membangun data base atau kebutuhannya menggunakan alat bantu. Pengguna ini dibedakan menjadi

a) DBMS system designers dan implementer, ialah pengguna yang merancang dan mengimplementasikan modul-modul dan interface menggunakan paket-paket software DBMS. (seperti:. Modul: catalog, *procs query lang., procs interface, access & buffering data, controlling cuncurrency, handling data recovery & security; interfacing: interface for integrated system*).

b) *Tool developers*. Pengguna yang merancang dan mengimplementasikan tools untuk mendukung software DBMS. Seperti Tools untuk meningkatkan performance database, tool untuk monitoring operasional database.

c) *Operators dan maintenance personnel*. Para personel administrator yang bertanggung jawab akan jalannyaoperasional database termasuk maintenance (hardware/software) DBMS.

**6) Operasi-Operasi dasar manajemen basis data**

Operasi-operasi dasar yang dapat kita lakukan berkenaan dengan basis data adalah sebagai berikut:

1. Pembuatan basis data baru (*create database*), adalah proses yang identik dengan pembuatan lemari arsip yang baru.
2. Penghapusan basis data (*drop database*), adalah proses yang identik dengan perusakan lemari arsip, sekaligus beserta isinya jika ada.

3. Pembuatan table baru ke suatu basis data (*create table*), yang identik dengan penambahan map arsip baru ke sebuah lemari arsip yang telah ada.
4. Penghapusan table dari suatu basis data (*drop table*), identik dengan perusakan map arsip lama yang ada di sebuah lemari arsip.
5. Penambahan / pengisian data baru di sebuah basis data (*insert)*, identik dengan penambahan lembaran arsip ke sebuah map arsip.
6. Pengambilan data dari sebuah table (*retrieve / search*), identik dengan pencarian lembaran arsip dalam sebuah map arsip.
7. Pengubahan data dalam sebuah table (*update*), identik dengan perbaikan isi lembaran arsip yang ada di sebuah map arsip.
8. Penghapusan data dari sebuah table (*delete*), identik dengan penghapusan sebuah lembaran arsip yang ada di sebuah map arsip.

**7) Pengenalan File tabel record dan field**

Didalam manajemen basis data, data disimpan dalam bentuk Berkas atau *file*. Berkas adalah himpunan seluruh record data (sisi baris) yang bertipe sama Suatu tabel atau Entitis dalam basis data relasional digunakan untuk mendukung antar muka komunikasi antara pemakai dengan para profesional komputer. Gambar dibawah ini menjelaskan contoh penempatan data mahasiswa dalam tabel MHS.

| NIM | NAMA | ALAMAT | KOTA | JURUSAN |
|---|---|---|---|---|
| 99.1102 | Sablah | Jl. Perak Timur 3 | Surabaya | Teknik Elektro |
| 99.1105 | Maimunah | Jl. Ketintang 17 | Surabaya | Teknik Kimia |
| 99.1113 | Abunawas | Jl. Blimbing II/25 | Malang | Teknik Informatika |

Gambar 2. Data-data pada Tabel MHS

Record atau Baris atau dalam istilah model relasional yang formal disebut dengan Tuple adalah kumpulan data yang terdiri dari satu atau lebih suatu field. Pada setiap baris-baris ini tersimpan data-data dari subyek tabel yang bersangkutan . Di samping itu data-data yang ada dalam satu record bias terdiri

dari bermacam-macam tipe data (Penjelasan tentang tipe dat kana dijelaskan pada bab selanjutnya). Contoh bentuk data yang terletak dalam satu record diperlihatkan dengan latar belakang hitam, seperti pada gambar dibawah ini.

MHS : Table

| NIM | NAMA | ALAMAT | KOTA | JURUSAN |
|---|---|---|---|---|
| 99.1102 | Sablah | Jl. Perak Timur 3 | Surabaya | Teknik Elektro |
| 99.1105 | Maimunah | Jl. Ketintang 17 | Surabaya | Teknik Kimia |
| 99.1113 | Abunawas | Jl. Blimbing II/25 | Malang | Teknik Informatika |

Record: 2 of 3

Gambar 3. Data-data pada satu record di tabel MHS

Field atau Kolom atau dalam istilah model relasional yang formla disebut dengan Attribute adalah kumpulan data yang mempunyai/menyimpan yang sama/sejenis untuk setiap pada tabel. Ynag perlu diperhatikan bahwa urutan data (fisiknya) dalam suatu kolom untuk tiap-tiap baris tidak memiliki arti sehingga data-data tersebut tidak berpengaruh walaupun diubah. Contoh bentuk data yang terletak pada satu field/kolom diperlihatkan dengan latar belakang hitam, seperti pada gambar dibawah ini

MHS : Table

| NIM | NAMA | ALAMAT | KOTA | JURUSAN |
|---|---|---|---|---|
| 99.1102 | Sablah | Jl. Perak Timur 3 | Surabaya | Teknik Elektro |
| 99.1105 | Maimunah | Jl. Ketintang 17 | Surabaya | Teknik Kimia |
| 99.1113 | Abunawas | Jl. Blimbing II/25 | Malang | Teknik Informatika |

Record: 1 of 3

Gambar 4. Data-data pada satu field NAMA di tabel MHS

## c. Rangkuman

Secara kharfiah, basis data terdiri dari dua kata yaitu basis dan data. Basis dapat diartikan sebagai suatu markas atau gudang, tempat bersarang atau tempat berkumpul.Data merupakan representasi dari fakta dunia(manusia, barang, peristiwa, keadaan)Secara istilah basis data adalah merupakan Kumpulan berkas atau tabel atau arsip yang saling berhubungan yang disimpan dalam media penyimpanan tertentu, dapat berupa media cetak maupun media

elektronik. Kompoenen basis data meliputi sistem komputer: hardware dan software, basisdata lain dan pengguna. Software meliputi sistem operasi, aplikasi pemrograman dan DBMS. DBMS merupakan gabungan antara basis data dan kumpulan program atau perangkat lunak DBMS (d*atabase management system) yaitu* program aplikasi yang dibuat dan bekerja dalam satu system.

Beberapa tujuan penggunaan basis data adalah berkaiatan dengan: 1) Kecepatan dan Kemudahan (*Speed*). 2) Efisiensi Ruang Penyimpanan (*Space*). 3) Keakuratan (*Accuracy*),4) Ketersediaan (*Availability*). 5) Kelengkapan (*Completeness*).6) Keamanan (*Security*) dan 7) Kebersamaan (*Sharability*). Sementara itu jenis-jenis pengguna basis data antara lain ialah : 1) Database Administrator. 2) *Database Designer. 3) Application Programmer. 4) End user. 5) System Analyst. 6) Worker behind the scene. Operasi-operasi yang dapat dilakukan dalam basis data antara lain ialah : 1) create database. 2) drop database.3) create table. 4) drop table. 5) insert data. 6) retrieve / search data. 7) update data dan 8) delete data.*

**d. Tugas : Mengoperasikan Aplikasi basis data**

Sebelum mengerjakan tugas, buatlah kelompok terdiri atas 2-3 orang. Dalam kegiatan ini peserta didik akan mengamati uraian materi konsep basis data dan mengoperasikan contoh aplikasi basis data. Contoh aplikasi basis data disediakan oleh guru atau teknisi.

1. Jalankan contoh aplikasi basis data yang telah disediakan. Amatilah layanan atau operasi-operasi basis data yang disediakan oleh aplikasi tersebut.
2. Jalankan atau lakukan operasi tambah data (insert data) dengan beberapa data yang berbeda. Amati perubahan yang terjadi.
3. Jalankan atau lakukan operasi update data terhadap data yang telah dimasukkan. Amati perubahan yang terjadi.
4. Jalankan atau lakukan operasi pencarian terhadap suatu data. Amati perubahan yang terjadi.
5. Jalankan atau lakukan operasi delete data terhadap suatu data yang telah dipilih. Amati perubahan yang terjadi.
6. Jika tersedia Jalankan atau lakukan operasi untuk membuat laporan (create report) yang siap dicetak oleh printer.

7. Jalankan atau lakukan operasi-operasi basis data lainnya yang tersedia dalam aplikasi tersebut. Amati perubahan yang terjadi.
8. Tuliskan dan kumpulkan data-data yang ada untuk setiap langkah diatas, kemudian analisis hasilnya menggunakan analisa diskriptif.
9. Diskusi dan komunikasikan hasil analisis dalam kelompok dan buatlah kesimpulan.
10. Buatlah laporan hasil eksperimen dan komunikasikan hasil laporan dengan guru pembimbing.

**e. Test Formatif.**

Dalam test ini setiap peserta didik membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

1. Jelaskan beberapa pengertian atau definisi basis data secara istilah?
2. Jelaskan pengertian Sistem manajemn basis data ?
3. Sebutkan dan jelaskan tujuan atau manfaat penggunaan basis data.?
4. Sebutkan dan jelaskan operasi-operasi dasar dalam basis data ?.

**f. Lembar Jawaban Test Formatif (LJ).**
**LJ- 01 : Pengertian atau definisi basis data.**

.............................................................................................................
.............................................................................................................
.............................................................................................................
.............................................................................................................
.............................................................................................................
.............................................................................................................
.............................................................................................................
.............................................................................................................
.............................................................................................................
.............................................................................................................
.............................................................................................................
.............................................................................................................
.............................................................................................................
.............................................................................................................

## Kegiatan belajar 2: Struktur hirarki Basis Data

**a. Tujuan Pembelajaran.**

Setelah mengikuti kegiatan belajar 2 ini diharapkan peserta didik dapat:

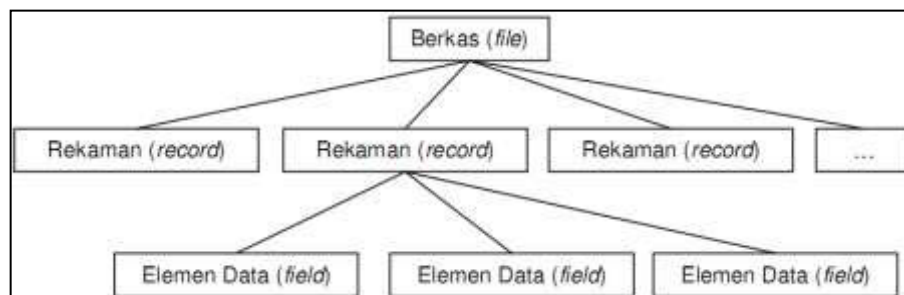
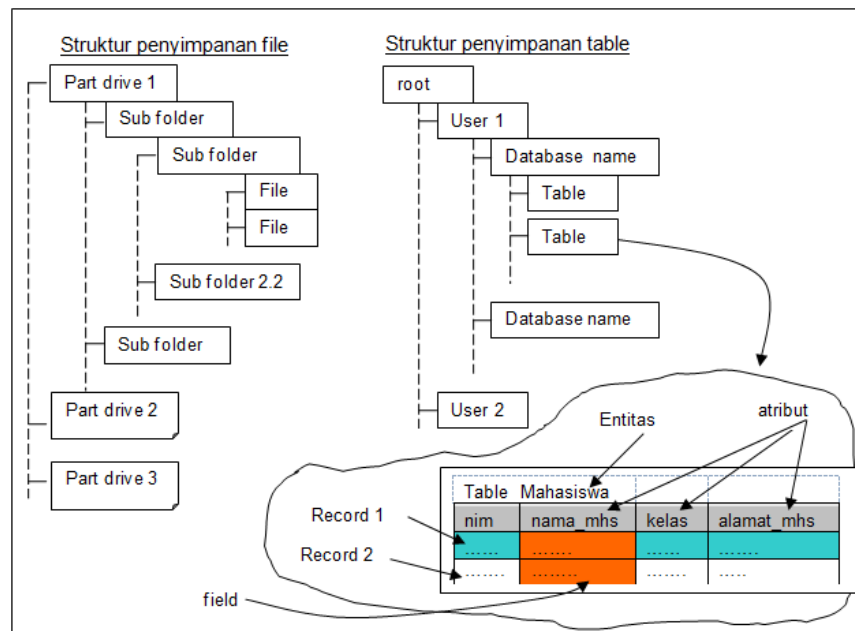
- ✓ Memahami konsep struktur dan hirarki basis data
- ✓ Membuat struktur hirarki aplikasi basis data.

**b. Uraian materi.**

**1) Definisi Struktur atau arsitektur Basis Data**

Arsitektur basis data merupakan serangkaian pengetahuan tentang pemodelan data. Pengetahuan tentang File, table, field, record indeks, abstraksi data dan serangkaian konsep yang digunakan untuk membuat diskripsi struktur basis data. Melalui diskripsi Struktur basis data dapat ditentukan jenis data, hubungan dan konstrain (keterbatasan) data yang ditangani. Dalam basis data, data diorganisasikan kedalam bentuk elemen data (field), rekaman (record), dan berkas (file). Definisi dari ketiganya adalah sebagai berikut:

- Elemen (kolom atau field) data adalah satuan data terkecil yang tidak dapat dipecah lagi menjadi unit lain yang bermakna. Misalnya data siswa terdiri dari NIS, Nama, Alamat, Telepon atau Jenis Kelamin.
- Rekaman (record) merupakan gabungan sejumlah elemen data yang saling terkait. Istilah lain dari record adalah baris atau tupel.
- Berkas(file) adalah himpunan seluruh record yang bertipe sama

Struktur hirarki sebuah database dapat digambarkan dalam diagram hirarki begai berikut :

Gambar 5. Struktur hirarki sistem basis data

Gambar 6. Struktur penyimpanan file dan tabel dalam basis data

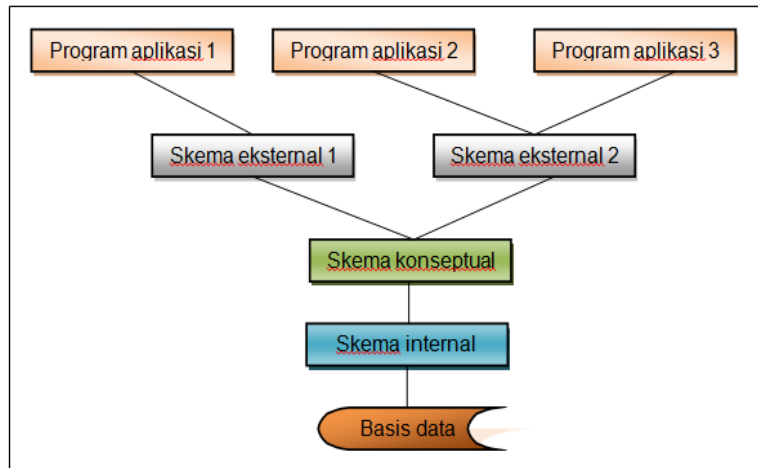
**2) Skema Atau Abstraksi Basis Data**

Abstraksi data adalah merupakan tingkatan atau level bagaimana melihat data dalam sistem basis data.Abstraksi data diwujudkan dalam pemodelan data yang merupakan sejumlah konsep yang digunakan untuk membuat diskripsi struktur basis data.Melalui diskripsi struktur basis data, dapat ditentukan jenis data dan hubungannya deangan data lain

Skema basis data merupakan diskripsi dari basis data yang spesifikasinya ditentukan dalam tahap perancangan. Skema ini digunakan untuk memisahkan antara fisik basis data dan program aplikasi pemakai. Penggambaran skema bsisi data biasanya ditampilkan dalam diagram yang berisi sebagian detail data dari diskripsi basis data. Secara umum arsitektur basis data menggunakan arsitektur tiga skema yang meliputi tiga level yaitu :

1. Level Internal atau skema internal. Level ini mendifinisikan secara detail penyimpanan basis data dan pengaksesan data. Pada level ini memuat diskripsi struktur penyimpanan basis data, menggunakan model data fisikal,
2. Level Konseptual (skema konseptual), memuat diskripsi struktur basis data seara keseluruhan untuk semua pemakai. Level ini memuat diskripsi

tentang entity, atribut, relasi dan konstrain tanpa memuat diskripsi data secara detail.

3. Level eksternal (skema eksternal atau view), mendefinisikan pandangan data terhadap sekelompok pemakai(local view) dengan menyembunyikan data lain yang tidak diperlukan oleh kelompok pemakai tersebut.

Gambar 7. Arsitektur tiga-skema sistem manajemen basis data

**3) Pemodelan data**

Pemodelan data merupakan sarana untuk melakukan abstraksi data. Merupakan sejumlah konsep untuk membuat diskripsi stuktur basis data. Kebanyakan model data memuat spesifikasi untuk operasi dasar (*basic operation)* dalam pengaksesan dan pembaharuan data. Pada perkembangan terakhir dikenal dengan istilah tabiat data (*data behavior*) pada pemrograman berorientasi object. Terdapat sejumlah cara dalam merepresentasikan model dalam perancangan basis data. Secara umum pemodelan data dapat dikelompokkan menjadi dua yaitu :

1. *Object based logical model*. Dalam pemodelan ini struktur atau hirarki basis data diilustrasikan berdasarkan object. Model ini meliputi: 1) Model keterhubungan entitas (Entity Relationalship Model atau ERD). 2) Model berorientasi object (Object-Oriented Model). 3) Model Data Semantik(Semantic Data Model). 2) Model data Fungsional (Function Data Model).
2. *Record-based logical model*. Dalam model ini struktur basis data diilustrasikan berdasarkan record. Model ini meliputi: 1) Model relational

(*Relational Model*). 2) Model Herarkis (Hierarchical Model) 3) Model Jaringan (*Network Model).*

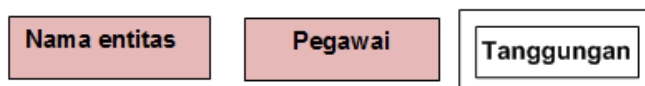
**4) Struktur konseptual basis data**

Tiga konsep dasar dalam pembuatan diskripsi struktur basis data yaitu model data konseptual, model data fisikal dan model view.Konseptual data menyajikan konsep tentang bagaimana user basis data memandang atau memberlakukan data. Konseptual merupakan level tinggi (hight level) yang dekat dengan user. Didalam Konseptual data menjelaskan beberapa hal yaitu entitas, atribute, key dan relasi antar entitas (akan dibawah dalam kegiatan belajar 3)

a) Entity atau Entitas

Entitas adalah obyek yang mewakili sesuatu dalam dunia nyata dan dapat dibedakan antara satu dengan lainnya (unique).Setiap entitas memiliki beberapa atribut yang mendeskripsikan karakteristik dari objek. Entitas dapat berupa:

- Data Fisik (seperti mobil, rumah, manusia, pegawai, peserta didik.
- Abstrak atau konsep (seperti department, pekerjaan, mata pelajaran)
- Kejadian (pembelian, penjualan, peminjaman, dll)

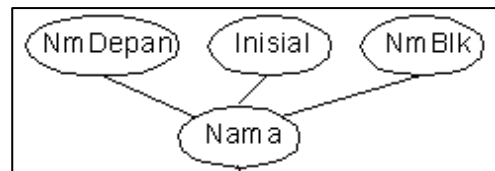
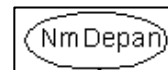
Entitas dapat dibedakan menjadi dua macam yaitu Entitas kuat dan entitas lemah. Entitas lemah adalah yang keberadaannya tergantung pada entitas lain. Gambar dibawah ini menjelaskan notasi umum entitas kuat dengan nama entitas pegawan dan entitas lemah dengan nama entitas tanggungan. Entitas tanggungan disebut sebagai entitas lemah karena jika data seorang pegawai dihapus maka data tanggungannya juga akan terhapus. Keberadaan data tanggungan tergantung pada data di pegawai

Gambar 8. Nnotasi entitas kuat (kotak satu) dan entitas lemah kotak dua

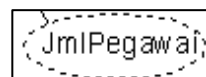
b) Atribute,

*Attribute* merupakan karakteristik dari entitas atau relationship, yang menyediakan penjelasan detail tentang entitas atau relationship. Dalam penerapannya (level fisik) atribut merupakan field atau kolom dari sebuah tabel. Misalnya entitas mahasiswa memiliki atribute nama, alamat, NIM. Berdasarkan

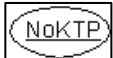
karakteristik sifatnya, atribut dapat dikelompokkan menjadi; 1) Simple attribute dan composite attribute. 2) Single valued attribute dan multi valued attribute. 3) Mandatory attribute 4) Derived attribute (attribut turunan) dan 5) key attribute.

Simple Attribute atau atomic attribute adalah attribut terkecil yang tidak bisa dipilah lagi. suatu atribut yang tidak dapat dibagi-bagi lagi menjadi atribut yang lebih kecil. Contohnya adalah atribut JenisKel pada entitas pegawai. Gambar diatas menjelaskan simbol atau notasi Simple Attribute

Composite attribute adalah atribut yang dapat dibagi menjadi atribut yang lebih kecil. Attribut ini dapat diartikan attribute atomic yang menggambarkan atribut dasar dengan suatu arti tertentu. Contoh: atribut Nama pada entitas pegawai dapat dipecah menjadi atribut NmDepan, Inisial dan NmBlk. Gambar diatasmenjelaskan simbol atau notasi *composite attribute*. Atribut nama merupakan *composite attribute.*

Single value Attribute adalah suatu atribut yang hanya mempunyai satu nilai. Misalnya atribut NmDepan pada entitas pegawai. NmDepan seorang pegawai selalu bernilai satu nilai, tidak mungkin lebih dari satu. Gambar diaasmenjelaskan simbol atau notasi Single value Attribute

Multi Value attribute adalah atribut yang dapat memiliki lebih dari satu nilai yang jenisnya sama dari sebuah data tunggal. Misalnya atribut lokasi pada entitas departemen dapat berisi 2 nilai atau lebih seperti Surabaya atau Jakarta. Gambar diatas menjelaskan simbol atau notasi Multi Value attribute

Derived Attribute atau Attribut Turunanadalah atribut yang nilai-nilainya diperoleh dari pengolahan atau dapat diturunkan dari atribut atau tabel lain yang berhubungan. Misalnyaatribut JmlPegawai pada entitas Departemen. Gambar diatas menjelaskan simbol atau notasi Multi Value attribute

c) Key attribute.

*Key* adalah merupakan suatu atribut yang menandakan kunci dari suatu entitas yang bersifat unik. Key attribute adalah satu atau beberapa atribut yang mempunyai nilai unik sehingga dapat digunakan untuk membedakan data pada suatu baris/record dengan baris lain pada suatu entitas. Key attribute dibedakan menjadi tiga yaitu: 1) *Superkey* 2) *Candidat Key* dan 3) *Primary* key

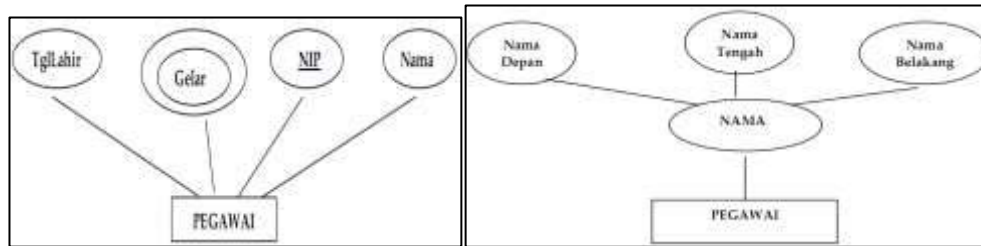
Tabel dibawah ini menjelaskan beberapa contoh nama entitas beserta nama atribut-atributnya

Tabel 1. Daftar entitas dan atributnya

| Nama entitas | Nama Atribute |
|---|---|
| Pegawai | NIP, NUPTK, Nama, Alamat, Agama, jenis kelamin |
| Siswa | NIS, Nama, Alamat, Agama, jenis kelamin |
| Mata pelajaran | Kode_mapel, Nama_mapel, Semester, |
| Departemen | No, Nama, lokasi |

Superkey adalah satu atau gabungan beberapa atribut yang dapat membedakan setiap baris data dalam sebuah tabel secara unik. Misalnya superkey untuk entitas pegawai antara lain: 1) NoKTP, Nama, Alamat, JenisKel, Gaji. 2) NoKTP, Nama, Alamat, JenisKel. 3) NoKTP, Nama, Alamat. 4) NoKTP, Nama. 5) Nama (jika dapat dijamin kalau tidak ada nama yang sama antara satu baris dengan baris yang lain). 6) NoKTP

Candidat Key adalah merupakan superkey yang jumlah atributnya paling sedikit. Misalnya candidat key untuk entitas pegawai antara lain:

- Nama (jika dapat dijamin kalau tidak ada nama yang sama antara satu baris dengan baris yang lain)
- NoKTP

Primary key adalah suatu candidat key yang dipilih menjadi kunci utama karena sering dijadikan acuan untuk mencari informasi, ringkas, menjadi keunikan suatu baris. Misalnya NoKTP antara satu pegawai dengan pegawai lain pasti berbeda, dalam hal ini noKTP dapat digunakan sebagai suatu key. Gambar diatas menjelaskan simbol atau notasi *primary key.*

Gambar 9. Contoh model struktur entitas pegawai

**5) Struktur Fisik Basis Data**

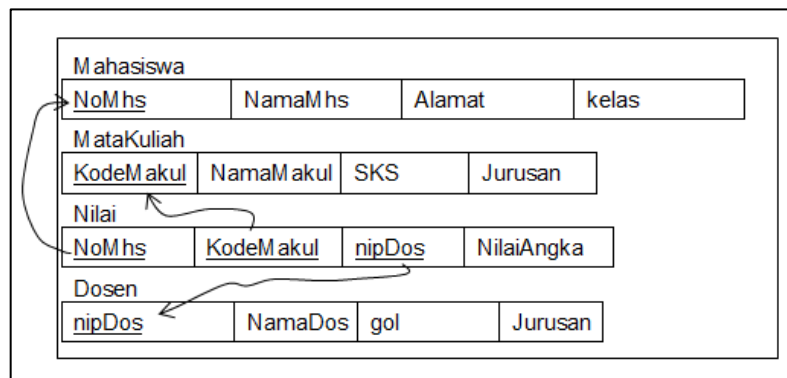
*Physical* data merupakan suatu konsep bagaimana diskripsi detail data disimpan dalam sebuah komputer. Physical data merupakan level rendah (low level) yang mendekati ke data sebenarnya. Dalam physical data menjelaskan definisi data yang meliputi nama atribut, type data (misalnya varchar, integer dll), size atau ukurannya data. Data yang diimplementasikan berupa table yang terdiri dari barisan data dalam kolom (field) dan baris (record). Setiap DBMS mempunyai aturan-aturan tersendiri dalam membuat definisi, struktur basis data dan tipe data yang digunakan.

Tabel 2. Jenis jenis tipe data dalam DBMS Microsoft access

| TIPE DATA | KETERANGAN |
|---|---|
| Text | Digunakan untuk field alfanumeric (misalnya nama, alamat, kode pos), memiliki banyak karakter yaitu maksimal 255 karakter pada setiap fieldnya. |
| Memo | Sama seperti text, tetapi dapat menampung kurang lebih 64.000 karakter untuk tiap fieldnya, tapi tidak bisa diurutkan/diindekskan. |
| AutoNumber | Tidak dapat disi secara manual tapi terisi secara otomatis oleh Access, secara berurutan atau acak biasanya digunakan untuk penomoran. |
| Number | Dapat digunakan untuk menyimpan data numeric yang akan digunakan untuk proses perhitungan matematis (mengurangi, menambahkan, mengkali dan membagi) suatu bilangan |
| Date/Time | Digunakan untuk data yang berjenis tanggal, waktu atau penggabungan dari tanggal dan waktu |
| Currency | Tipe jenis number, tetapi pada awal angka selalu disertakan symbol currency default sesuai dengan regional setting yang digunakan, misalnya RP. $. Dapat menggunakan angka dengan 15 dgiit dibelakang desimal dan 4 digit sesudah desimal |

| Yes/No | Merupakan tipe data dengan 2 pilihan saja yaitu Yes (1 atau Ture) dan No (0 atau False). Format yang tersedia adalah Yes/No, True/False, dan On/OFF |
|---|---|
| OLE Object | Digunakan untuk eksternal objek, seperti bitmap atau file suara |
| Hyperlink | Digunakan untuk menyimpan alamat internet atau file yang ditunjukan melalui alamat URL |
| Lookup Wizard | Jika menggunakan tipe data ini untuk sebuah field, maka bisa memilih sebuah nilai dari tabel lain atau dari sebuah daftar nilai yang ditampilkan dalam combobox |

| Table Mahasiswa: | Tabel Dosen: |
|---|---|
| ¬ NoMhs; Number (10); primary key,<br>¬ NamaMhs, Text(30),<br>¬ Alamat; Text (50),<br>¬ Kelas; char (10),<br>¬ MhsFoto; OLE Object | ¬ nipDos; Number (20); primary key,<br>¬ NamaDos, Text(30),<br>¬ Gol; Text (5),<br>¬ Jurusan; Tex(10),<br>¬ dosFoto; OLE Object |

Gambar 10. Contoh diskripsi struktur tabel mahasiswa dan tabel dosen

Gambar 11. Contoh struktur tabel dalam basis data sistem nilai mahasisw

## c. Rangkuman

Struktur atau arsitektur basis data merupakan serangkaian pengetahuan tentang komponen penyusun data beserta hubungan komponen tersebut. Representasi struktur basis data diwujudkan dalam pemodelan data. Struktur tersebut meliputi File, table, field, record indeks, abstraksi data dan serangkaian konsep yang digunakan untuk membuat diskripsi struktur basis data. Abstraksi data merupakan suatu pendekatan dalam menggambarkan suatu data. Abstraksi data dapat diwujudkan dalam suatu skema basis data. Skema basis data

merupakan diskripsi dari basis data yang spesifikasinya ditentukan dalam tahap perancangan. Skema ini digunakan untuk memisahkan antara fisik basis data dan program aplikasi pemakai.

Arsitektur yang sering digunakan untuk membuat abstraksi data adalah arsitektur tiga skema yang meliputi tiga level yaitu: 1) Level Internal atau skema internal. 2) Level Konseptual atau skema konseptual 3) Level eksternal (skema eksternal atau view). pemodelan data dapat dikelompokkan menjadi dua yaitu : 1) *Object based logical model* dan 2) *Record-based logical model.*

Skema atau level Konseptual data menjelaskan tentang entitas, atribute, key dan relasi antar entitas. Entitas adalah obyek yang mewakili sesuatu dalam dunia nyata dan dapat dibedakan antara satu dengan lainnya (unique). *Attribute* merupakan karakteristik dari entitas atau relationship. *Key* adalah merupakan suatu atribut yang menandakan kunci dari suatu entitas yang bersifat unik

*Physical* data merupakan suatu konsep bagaimana diskripsi detail data disimpan dalam sebuah komputer. Physical data menjelaskan definisi data yang meliputi nama atribut, type data (misalnya varchar, integer dll), size atau ukurannya data. Setiap DBMS mempunyai aturan-aturan tersendiri dalam membuat definisi, struktur basis data dan tipe data yang digunakan..

**d. Tugas : Mengoperasikan Aplikasi basis data**

Dalam kegiatan ini peserta didik akan melakukan eksperimen atau praktikum secara berkelompok satu kelompok terdiri dari dua sampai tiga orang. Eksperimen dilakukan melalui pengamatan terhadap contoh aplikasi pada kegiatan 1 kemudian merancang dan membuat struktur basis datanya. Bacalah seluruh langkah eksperimen dibawah ini kemudian lakukan dengan cermat dan teliti dengan perangkat yang telah disediakan.

1. Jalankan contoh aplikasi basis data yang telah disediakan. Amatilah tabel-tabel yang ada dalam aplikasi database tersebut.
2. Berdasarkan pengamatan pada langkah 1, identifikasi entitas-entitas basis data dan tulislah diskripsi singkat tentang entitas tersebut.
3. Untuk setiap entitas tambahkan atribute-atribute yang ada, tulislah dalam bentuk tabel dan tentukan pula atribute key (primary key).

4. Dengan menggunakan notasi yang telah dijelaskan gambarkan struktur basis data level konseptual yang menjelaskan entitas beserta atribut-atributnya, tanpa mnggambarkan relasi antar entitas.
5. Dari gambar diagram struktur entitas pada langkah 4, buatlah peta pengkodean record data (struktur level fisik).
6. Dengan merujuk DBMS micosoft access buatlah diskripsi setiap tabel dalam gambar langkah 5. Untuk setiap atribut tentukan tipe data, ukuran data dan key atribut (primary key).
7. Kumpulkan data-data setiap langkah dan analisis data tersebut menggunakan analisis diskriptif.
8. Diskusi dan komunikasikan hasilnya dalam kelompok dan buatlah kesimpulan.
9. Buatlah Laporan dan komunikasikan hasil laporan dan pembahasan dengan guru pembimbing

**e. Test Formatif.**

Dalam test ini setiap peserta didik membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

1. Jelaskan secara singkat definisi struktur atau arsitektur basis data ?
2. Jelaskan, gambarkan pengertian arsitektur tiga skema basis data?
3. Jelaskan secara singkat dan berikan contoh pengertian entitas, atribut dan key atribut ?
4. Jelaskan secara singkat pengertian struktur fisik basis data ?

**f. Lembar Jawaban Test Formatif (LJ).**

**LJ- 01 : Pengertian Struktur atau arsitektur basis data.**

.....................................................................................................

.....................................................................................................

.....................................................................................................

.....................................................................................................

.....................................................................................................

.....................................................................................................

.....................................................................................................

.....................................................................................................

g. **Lembar Kerja Peserta Didik.**

## Kegiatan belajar 3: ERD- Identifikasi Entitas dan Atribut

**a. Tujuan Pembelajaran.**

Setelah mengikuti kegiatan belajar 3 ini diharapkan peserta didik dapat:

- ✓ Memahami konsep entitas atau *entity-relationship diagram* (ERD)
- ✓ Mendefinisikan diskripsi sistem basis data (role of bisnis)
- ✓ Mengidentifikasi entitas sistem basis data
- ✓ Mengidentifikasi atribut sistem basis data.
- ✓ Membuat struktur entitas beserta atributnya..

**b. Uraian materi.**

**1) Definisi ERD**

Diagram relasi entitas atau *entity-relationship diagram* (ERD) adalah suatu diagram dalam bentuk gambar atau simbol yang mengidentifikasi tipe dari entitas di dalam suatu sistem yang diuraikan dalam data dengan atributnya, dan menjelaskan hubungan atau relasi diantara entitas tersebut. ERD merupakan model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. ERD berupa model data konseptual, yang merepresentasikan data dalam suatu organisasi.ERD menekankan pada struktur dan relationship data. ER diagram digunakan oleh profesional sistem untuk berkomunikasi dengan pemakai eksekutif tingkat tinggi dalam perusahaan atau organisasi yang tidak tertarik pada pelaksanaan operasi sistem sehari-hari, namun lebih menekankan kepada beberapa hal yaitu :

- Data apa saja yang diperlukan untuk bisnis mereka?
- Bagaimana data tersebut berelasi dengan data lainnya?
- Siapa saja yang diperbolehkan mengakses data tsb?

Untuk menggambarkan ER diagaram setidaknya ada tiga langkah yang harus dilakukan oleh perancang basis data yaitu:

1. Menemukan atau mendefinisikan Entitas
2. Menemukan atau mendefinisikan atribute
3. Menemukan atau mendefinisikan Relasi
4. Menggambarkan ERD menggunakan notasi-notasi standar.

**2) Menemukan Entitas**

Sebagaimana telah dijelaskan secara lengkap dalam uraian materi kegiatan belajar 2, entitas adalah obyek yang mewakili sesuatu dalam dunia nyata dan dapat dibedakan antara satu dengan lainnya *(unique).*Setiap entitas memiliki beberapa atribut yang mendeskripsikan karakteristik dari objek tersebut.

Adapun langkah-langkah yang seharusnya dilakukan untukmenemukan atau mendefinisikan Entitas dalam suatu sistem data base adalah sebagai berikut :

1. Buat ilustrasi atau gambaran cerita (*role of bussiness*) tentang sistem yang akan dicari entitasnya.
2. Tandai setiap objek yang diwakili oleh kata benda yang ada di dalam ilustrasi tersebut.
3. Untuk setiap objek tersebut yakinkan bahwa ia memiliki karakteristik yang nanti disebut sebagai atribut.
4. Tentukan objek yang merupakan entitas (Jika memang ia memiliki karakteristik jadikan ia sebagai entitas)
5. Menggambarkan entitas beserta atributnya menggunakan notasi simbol yang telah ditentukan.

Contoh : Sistem data base Kepegawaian di perusahaan A

1. Langkah 1: Membuat gambaran cerita tentang sistem kepegawaian di suatu perusahaan A.

   Perusahaan A memiliki 100 pegawai. Setiap pegawai dipimpin pengawas/mandor dari pegawai perusahaan itu sendiri dan tidak semua pegawai memimpin pegawai yang lain. sehingga satu pengawas dapat memimpin beberapa pegawai. Setiap pegawai bekerja untuk suatu departemen dan dalam suatu departemen dapat terdiri dari beberapa pegawai. Setiap departemen dikepalai oleh seorang pegawai yang bekerja mulai tanggal tertentu. Sebuah departemen dapat berada di beberapa lokasi. Selain bekerja di suatu departemen pegawai dapat bekerja pada beberapa proyek. Setiap proyek dikendalikan/diatur oleh suatu departemen, namun suatu departemen tidak harus mengendalikan/mengatur proyek. Satu departemen dapat mengendalikan beberapa proyek dan satu proyek hanya

dikendalikan oleh satu departemen Satu proyek dapat terdiri dari beberapa pegawai. Untuk keperluan penggajian perusahaan memerlukan data tanggungan pegawai. Seorang pegawai dapat menanggung beberapa tanggungan. Jika seorang pegawai pindah maka datanya akan dipindahkan / dihapus berikut data tanggungan / keluarganya.

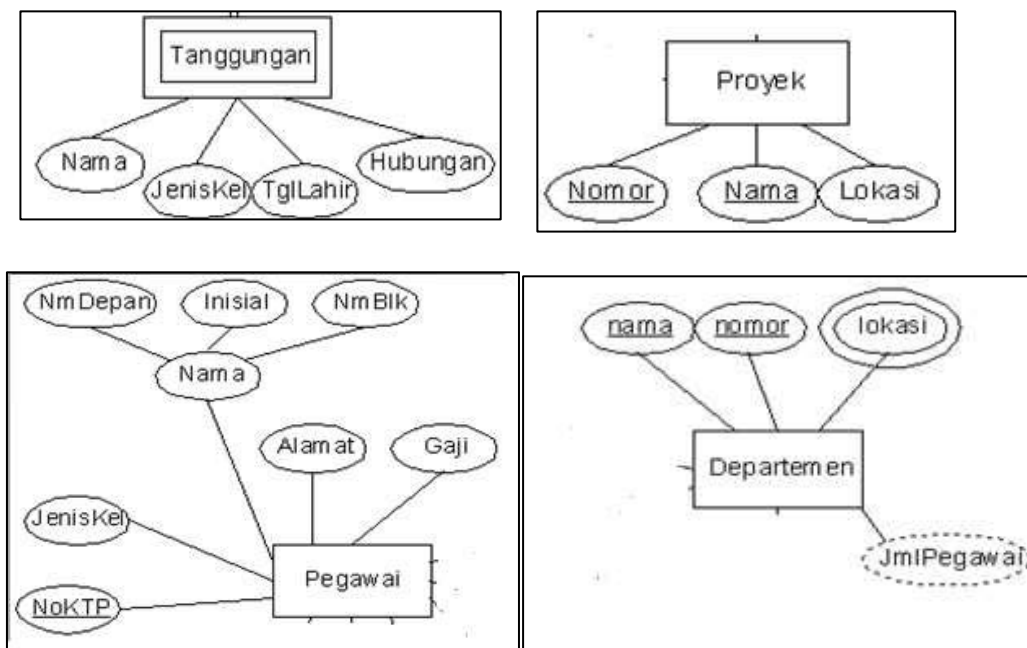
2. Langkah 2. Menandai pada soal cerita diatas setiap objek yang diwakili oleh kata benda yang ada di dalam ilustrasi tersebut
3. Langkah 3: Untuk setiap objek tersebut yakinkan bahwa ia memiliki karakteristik yang nanti disebut sebagai atribut. Sehingga kita menemukan entitas dan kemungkinan atributnya adalah sebagai berikut :
   a) Perusahaan: NoPerusahaan, nama, alamat
   b) Pegawai: NoKTP, Nama, Alamat, Jenis kelamin,gaji
   c) Pengawas:NoKTP, Nama, Alamat, Jenis kelamin,gaji
   d) Departemen: Nomor, Nama, lokasi, jumlah pegawai
   e) Lokasi : lokasi
   f) Proyek: Nomor, nama, lokasi
   g) Tanggungan: nama, jenis kelamin, tanggal lahir, hubungan dengan pegawai
4. Langkah 4: Tentukan objek yang merupakan entitas (Jika memang ia memiliki karakteristik jadikan ia sebagai entitas)
   a) Perusahaan: NoPerusahaan, nama, alamat (hanya berisi satu baris data) → bukan entitas
   b) Pegawai: NoKTP, Nama, Alamat, Jenis kelamin,gaji → entitas kuat
   c) Pengawas:NoKTP, Nama, Alamat, Jenis kelamin,gaji → sama dengan entitas Pegawai
   d) Departemen: Nomor, Nama, lokasi, jumlah pegawai → entitas kuat
   e) Lokasi : lokasi (karakteristiknya departemen, tidak memiliki karakteristik lain (unik)) → bukan entitas
   f) Proyek: Nomor, nama, lokasi → entitas kuat
   g) Tanggungan: nama, jenis kelamin, tanggal lahir, hubungan dengan pegawai→ merupakan entitas lemah karena keberadaannya tergantung dari entitas kuat pegawai.

### 3) Menemukan atribut.

Sebagaimana dijelaskan dalam uraian materi kegitan belajar 2, atribut adalah merupakan sifat-sifat atau karakteristik pada suatu entitas. Nama atribut ini identik dengan nama kolom atau field pada suatu tabel dalam basis data. Atribut dapat dibedakan menjadi beberapa macam antara lain adalah:

1. Simple Attribute dan Composite Attribute
2. Single Valued Attribute dan Multi Valued Attribute
3. Mandatory Attribute
4. Derived Attribute (Attribut Turunan)
5. Key Attribute (Atribut Kunci)

Adapaun untuk menemukan atribut dpat dilakukan melalui langkah-langkah dibawah ini yaitu :

1. Tentukan dan lengkapi karakteristik dari tiap-tiap entitas
2. Dari setiap karakteristik tersebut tentukan termasuk atribut apa
3. Gambarkan entitas beserta atributnya dengan notasi yang sesuai

Gambar 12. Diagram struktur entitas beserta atributnya.

**c. Rangkuman**

Diagram relasi entitas atau *entity-relationship diagram* (ERD) adalah suatu diagram dalam bentuk gambar atau simbol yang mengidentifikasi tipe dari entitas di dalam suatu sistem yang diuraikan dalam data dengan atributnya, dan menjelaskan hubungan atau relasi diantara entitas tersebut. Untuk menggambarkan ER diagaram setidaknya ada tiga langkah yang harus dilakukan oleh perancang basis data yaitu: 1) Menemukan atau mendefinisikan Entitas. 2) Menemukan atau mendefinisikan atribute. 3) Menemukan atau mendefinisikan Relasi. 4) Menggambarkan ERD menggunakan notasi-notasi standar

Langkah-langkah dilakukan untukmenemukan atau mendefinisikan Entitas yaitu: 1) membuat ilustrasi cerita (*role of bussiness*) sistem basis data. 2) menandai setiap objek yang diwakili oleh kata benda dari ilustrasi tersebut. 3) Untuk setiap objek atau entitas tersebut yakinkan bahwa telah memiliki karakteristik sebagai atribut. 4) menentukan objek yang merupakan entitas, Jika memiliki karakteristik maka menjadi sebuah entitas.

Adapun untuk menemukan atribut dapat dilakukan melalui langkah-langkah berikut yaitu : 1) Mentukan dan melengkapi karakteristik dari tiap-tiap entitas 2) Dari setiap karakteristik tersebut tentukan termasuk atribut apa. 3) Gambarkan entitas beserta atributnya dengan notasi yang sesuai.

**d. Tugas: Mengidentifikasi Entitas dan atribut**

Dalam kegiatan ini peserta didik akan melakukan eksperimen atau praktikum secara berkelompok satu kelompok terdiri dari dua sampai tiga orang. Dalam eksperimen ini peserta didik akan merancang diskripsi sistem basis data (role of bisnis) dan membuat struktur entitas beserta atributnya. Topik bisa ditentukan sendiri atau memilih beberapa alternatif seperti: basis data persewaan buku, mobil, DVD, Penjualan buku, ATK , komputer, HP, basis data kependudukan, pelatihan atau kursus, jasa perbaikan, mobil, barang elektonik dan lain-lain. Bacalah seluruh langkah eksperimen dibawah ini kemudian lakukan dengan cermat dan teliti instruksi yang diberikan:

1. Diskusikan dalam kelompok dan Tentukan topik atau judul sistem basis data yang akan dibuat, koordinasikan dan konsultasikan dengan guru atau teknisi.

2. Buat ilustrasi atau gambaran cerita (*role of bussiness*) tentang sistem basis data yang telah ditentukan.
3. Identifikasi atau temukan entitas dari diskripsi yang telah dibuat, dengan menandai (menggaris bawahi setiap objek yang diwakili oleh kata benda yang ada di dalam ilustrasi tersebut
4. Identifikasi atau tentukan dan lengkapi karakteristik dari tiap-tiap entitas dengan atribut-atribut dan key atribut (primery key).Tampilkan hasilnya dalam tabel.
5. Tentukan pula jenis atau tipe atribut-atributnya (sesuai dengan jenis atribut dalam uraian materi). Tampilkan hasilnya dalam tabel
6. Gambarkan entitas beserta atributnya dengan notasi yang sesuai.
7. Kumpulkan data-data setiap langkah dan analisis data tersebut menggunakan analisis diskriptif.
8. Diskusi dan komunikasikan hasilnya dalam kelompok dan buatlah kesimpulan.
9. Buatlah Laporan dan komunikasikan hasil laporan dan pembahasan dengan guru pembimbing.

**e. Test Formatif.**

Dalam test ini setiap peserta didik membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

1. Jelaskan secara singkat definisi ERD ?
2. Jelaskan secara singkat langkah-langkah untuk mengidentifikasi atau menemukan entitas ?
3. Jelaskan secara singkat langkah-langkah untuk mengidentifikasi atau menemukan atribut ?

**f. Lembar Jawaban Test Formatif (LJ).**

**LJ- 01 : Pengertian *entity relationship diagram* (ERD)**

...............................................................................................................

...............................................................................................................

...............................................................................................................

## Kegiatan belajar 4: ERD - Relasi Antar Entitas

**a. Tujuan Pembelajaran.**

Setelah mengikuti kegiatan belajar 4 ini diharapkan peserta didik dapat:

- ✓ Memahami konsep entitas atau *entity-relationship diagram* (ERD)
- ✓ Memahami batasan partisipasi atau constraint.
- ✓ Mengidentifikasi relasi dalam sistem basis data
- ✓ Membuat *entity-relationship diagram* (ERD).

**b. Uraian materi.**

**1) Definisi ERD**

Diagram relasi entitas atau *entity-relationship diagram* (ERD) adalah suatu diagram dalam bentuk gambar atau simbol yang mengidentifikasi tipe dari entitas di dalam suatu sistem yang diuraikan dalam data dengan atributnya, dan menjelaskan hubungan atau relasi diantara entitas tersebut. ERD merupakan model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. ERD berupa model data konseptual, yang merepresentasikan data dalam suatu organisasi.ERD menekankan pada struktur dan relationship data. ER diagram digunakan oleh profesional sistem untuk berkomunikasi dengan pemakai eksekutif tingkat tinggi dalam perusahaan atau organisasi yang tidak tertarik pada pelaksanaan operasi sistem sehari-hari, namun lebih menekankan kepada beberapa hal yaitu :

- Data apa saja yang diperlukan untuk bisnis mereka?
- Bagaimana data tersebut berelasi dengan data lainnya?
- Siapa saja yang diperbolehkan mengakses data tsb?

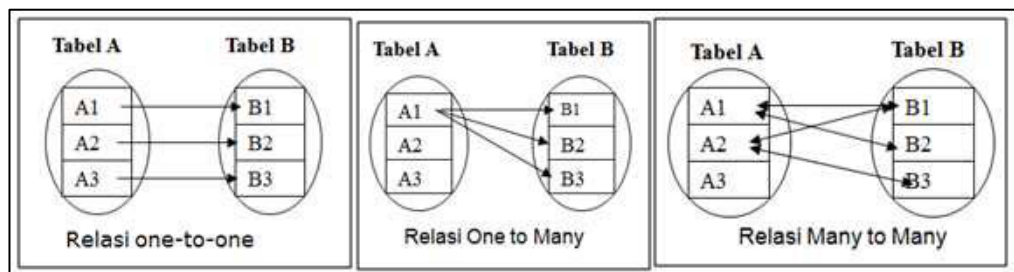
Untuk menggambarkan ER diagaram setidaknya ada tiga langkah yang harus dilakukan oleh perancang basis data yaitu:

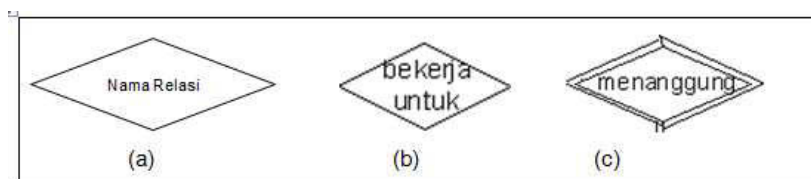
1. Menemukan atau mendefinisikan Entitas.
2. Menemukan atau mendefinisikan atribute.
3. Menemukan atau mendefinisikan Relasi.
4. Menggambarkan ERD menggunakan notasi-notasi standar.

**2) Relasi**

Relasi menyatakan hubungan antara dua atau beberapa entitas. Setiap relasi mempunyai batasan (*constraint*) terhadap kemungkinan kombinasi entitas yang berpartisipasi. Batasan tersebut ditentukan dari situasi yang diwakili relasi tersebut. Ragam atau jenis relasi dibedakan menjadi beberapa macam antara lain adalah :

1. Relasi *Binary*. Relasi binary merupakan relasi antara dua entitas. Relasi binary ini dibedakan menjadi :
   - Relasi One-to-one (notasi 1:1)
   - Relasi One-to-many (notasi 1:N) atau many-to-one (notasi N:1)
   - Relasi Many-to-many (notasi M:N)
2. *Relasi Ternary*. Relasi ternary adalah merupakan relasi antara tiga entitas atau lebih.

Dalam Relasi One-to-one (1:1) setiap atribute dari satu entitas berpasangan dengan satu attribute dari entitas yang direlasikan. Dalam relasi One-to-many (1:N) atau many-to-one (N:1) satu atribute berelasi dengan beberapa atribute dari entitas yang direlasikan. Dalam Many-to-many (M:N) satu atribute berelasi dengan beberapa atribute dari entitas yang direlasikan. Begitu pula sebaliknya.

Gambar 13. Ragam relasi antar entitas

Sebagaimana entias dalam relasi juga dapat dibedakan menjadi relasi kuat dan relasi lemah. gambar dibawah ini menjelaskan notasi umum untuk relasi kuat dan relasi lemah.

Gambar 14. Notasi relasi entitas untuk entitas kuat (b) dan entitas lemah (c)

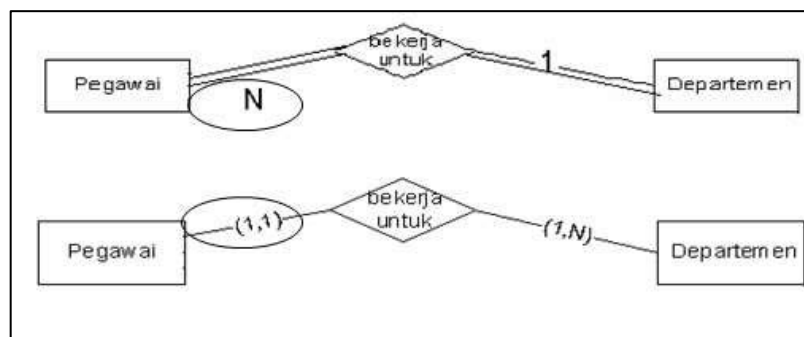
### 3) Batasan Partisipasi

Batasan partisipasi atau batasan hubungan entitas menjelaskan bagaimana data itu berelasi, batasan ini menentukan bagaimana (harus ataukah tidak) berpartisipasi suatu entitas dengan relasinya pada entitas lain. Batasan partisipasi dibedakan menjadi dua yaitu : 1) Partisipasi Total (harus berpartisipasi) dan 2) Partisipasi Parsial (tidak harus berpartisipasi)

Contoh relasi yang merupakan partisipasi total adalah relasi antara pegawai dengan departemen dengan nama relasi bekerja untuk dan partisipasi total disisi pegawai. Dari diskripsi basis data disebutkan bahwa :

"Semua pegawai harus bekerja di bawah suatu departemen"

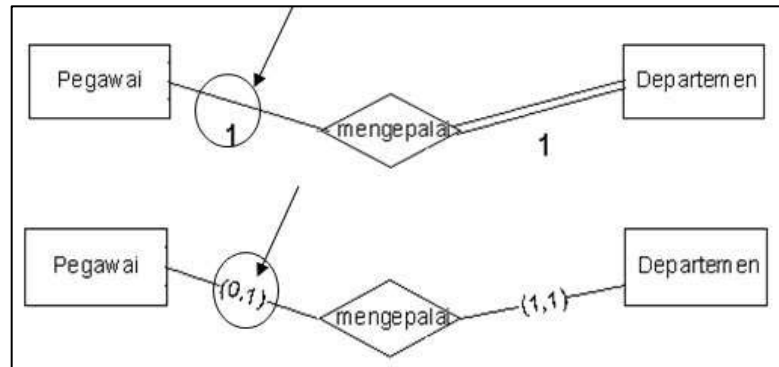
Dari pernyataan diatas mengindikasikan bahwa relasi disisi pegawai adalah relasi total yang ditandai dengan kata kunci harus. Untuk menggambarkan relasi dengan partisipasi total tersebut dapat dilakukan dengan dua pendekatan yaitu:

- Menggunakan garis ganda pada relasi disisi pegawai
- Menggunakan satu garis pada relasi disisi pegawai digabungkan dengan minimum 1 (minimum bekerja pada 1 departemen)

**Gambar 15. Relasi dengan batasan partisipasi total**

Contoh relasi yang merupakan partisipasi parsial adalah relasi antara pegawai dengan departemen dengan nama relasi mengepalai daan partisipasi parsial disisi pegawai. Dari diskripsi basis data disebutkan bahwa :

"Beberapa pegawai mengepalai sebuah departemen (setiap pegawai tidak harus mengepalai suatu departemen) "

Dari pernyataan diatas mengindikasikan bahwa relasi disisi pegawai adalah mempunyai partisipasi parsial. Hal ini ditandai dengan kata kunci (beberapa pegawai ...... atau tidak harus.....). Untuk menggambarkan relasi dengan partisipasi parsial tersebut dapat dilakukan dengan dua pendekatan yaitu:

- Menggunakan satu garis pada relasi disisi pegawai
- Menggunakan satu garis pada relasi disisi pegawai digabungkan dengan minimum 0 (tidak mengepalai departemen)

**Gambar 16. Relasi dengan batasan (constraint) partisipasi parsial**

#### 4) Menemukan Relasi.

Beberapa langkah yang dapat dilakukan untuk menemukan atau mengidentifikasi relasi yaitu antara lain sebagai berikut:

1. Dari gambaran cerita sistem, tandai setiap hubungan yang diwakili oleh kata kerja yang ada di dalam ilustrasi beserta entitas yang berhubungan
2. Identifikasikan rasio kardinalitas dari setiap hubungan
3. Identifikasikan batasan partisipasi dari setiap hubungan yang ada berikut kemungkinan atribut yang muncul dari setiap hubungan
4. Gambarkan hubungan tersebut dalam bentuk notasi diagram dan gabungkan dengan notasi Entitas dan atribut yang dibuat sebelumnya

Sebagai contoh adalah "Temukan relasi untuk Sistem Kepegawaian di perusahaan A dengan ( lihat kembali diskripsi sistem basis data diatas) ?"

Langkah-langkah penyelesaian adalah :

1. Langkah 1: dari gambaran cerita sistem, tandai dan tentukan setiap hubungan yang diwakili oleh kata kerja yang ada di dalam ilustrasi dan entitas yang berhubungan
2. Identifikasi hubungan antara entitas. Indentifikasi hubungan dilakukan dengan membuat tabel sepeti terlihat di bawah ini. Hubungan berlangsung dua arah dari entitas 1 ke entitsas 2 dan sebaliknya. Kata kunci hubungan satu sisi menggunakan kata aktif dan dari sisi sebaliknya menggunakan kata kunci pasif.

Tabel 4. Identifikasi hubungan antara dua entitas dua arah

| Entitas 1 | Hubungan | Entitas 2 |
|---|---|---|
| Pengawas (Pegawai) | memimpin | Pegawai |
| Pegawai | dipimpin | Pengawas(Pegawai) |
| Pegawai | bekerja untuk | Departemen |
| Departemen | terdiri dari | Pegawai |
| Pegawai | mengepalai | Departemen |
| Departemen | dikepalai | Pegawai |
| Pegawai | bekerja pada | Proyek |
| Proyek | terdiri dari | Pegawai |
| Departemen | mengatur | Proyek |
| Proyek | diatur | Departemen |
| Pegawai | menanggung | Tanggungan |
| Tanggungan | ditanggung | Pegawai |

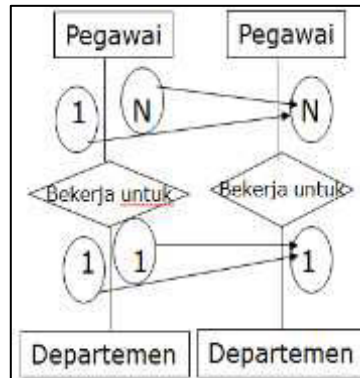
Tabel 5. Identifikasi hubungan antara dua entitas satu arah

| Entitas 1 | Hubungan | Entitas 2 |
|---|---|---|
| Pengawas(Pegawai) | memimpin | Pegawai |
| Pegawai | bekerja untuk | Departemen |
| Pegawai | mengepalai | Departemen |
| Pegawai | bekerja pada | Proyek |
| Departemen | mengatur | Proyek |
| Pegawai | menanggung | Tanggungan |

Tabel 6. Identifikasikan rasio kardinalitas dari setiap hubungan

| Entitas 1 | Banyaknya Entitas 1 yang berpartisipasi | Hubungan | Banyaknya Entitas 2 berpartisipasi | Entitas 2 |
|---|---|---|---|---|
| Pegawai | 1 | memimpin | N | Pegawai |
| Pegawai | 1 | dipimpin | 1 | Pegawai |
| Pegawai | 1 | bekerja untuk | 1 | Departemen |
| Departemen | 1 | terdiri dari | N | Pegawai |
| Pegawai | 1 | mengepalai | 1 | Departemen |
| Departemen | 1 | dikepalai | 1 | Pegawai |
| Pegawai | 1 | bekerja pada | N | Proyek |
| Proyek | 1 | terdiri dari | N | Pegawai |
| Departemen | 1 | mengatur | N | Proyek |
| Proyek | 1 | diatur | 1 | Departemen |
| Pegawai | 1 | menanggung | N | Tanggungan |

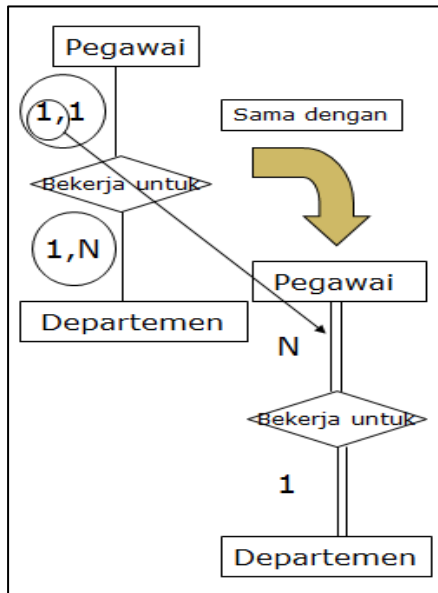
| Tanggungan | 1 | ditanggung | 1 | Pegawai |
|---|---|---|---|---|

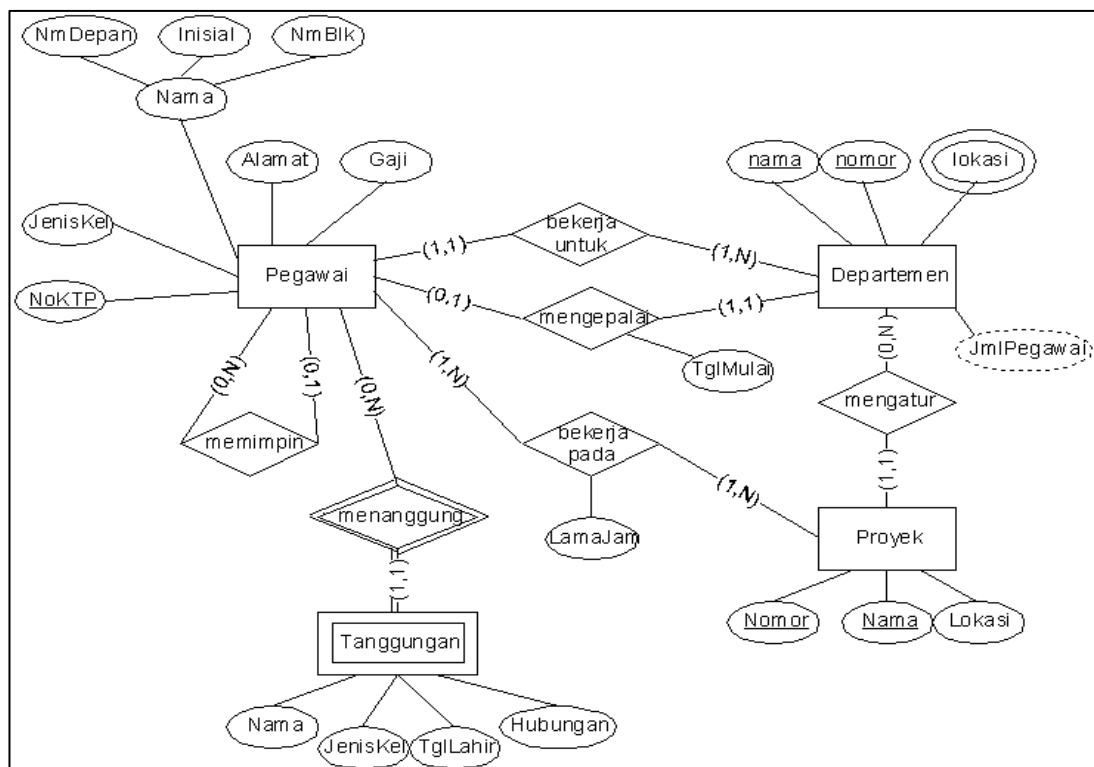
Dari tabel Identifikasikan rasio kardinalitas untuk setiap hubungan diatas dapat digambarkan diagram relasi antar entitas, seperti terlihat dalam gambar dibawah ini :

Gambar 17. Diagram relasi entitas pegawai dan departemen

Relasi antar entitas juga dapat diwujudkan dengan melibatkan identifikasikan batasan partisipasi dari setiap hubungan yang ada. Tabel dibawah ini emnejelaskn relasi yang melibatkan banyaknya partisipasi (minimal dan maksimal).

Tabel 7. Indentifikasi batasan partisipasi (min, max) antara dua entitas.

| **Entitas 1** | **Banyaknya Entitas 1 yang berpartisipasi** | **Hubungan** | **Banyaknya Entitas 2 yang berpartisipasi (min,max)** | **Entitas 2** |
|---|---|---|---|---|
| Pegawai | 1 | memimpin | (0,N) | Pegawai |
| Pegawai | 1 | dipimpin | (0,1) | Pegawai |
| Pegawai | 1 | bekerja untuk | (1,1) | Departemen |
| Departemen | 1 | terdiri dari | (1,N) | Pegawai |
| Pegawai | 1 | mengepalai | (0,1) | Departemen |
| Departemen | 1 | dikepalai | (1,1) | Pegawai |
| Pegawai | 1 | bekerja pada | (1,N) | Proyek |
| Proyek | 1 | terdiri dari | (1,N) | Pegawai |
| Departemen | 1 | mengatur | (0,N) | Proyek |
| Proyek | 1 | diatur | (1,1) | Departemen |

| Pegawai | 1 | menanggung | (0,N) | Tanggungan |
|---|---|---|---|---|
| Tanggungan | 1 | ditanggung | (1,1) | Pegawai |

Dari tabel indentifikasi batasan partisipasi (min, max) diatas dapat digambarkan diagram relasi entitasnya,seperti terlihat digambar samping.

Dengan cara yang sama dapat ditemukan digambarkan relasi entitas-entitasnya. Gambar dibawah ini menjelaskan ER diagram secara lengkap untuk sistem kepegawaian di perusahaan A

Gambar 18. Entity relationship diagram sistem basis data kepegawaian.

**c. Rangkuman**

Relasi menyatakan hubungan antara dua atau beberapa entitas. Setiap relasi mempunyai batasan (*constraint*) terhadap kemungkinan kombinasi entitas yang berpartisipasi. Batasan partisipasi atau batasan hubungan entitas menjelaskan bagaimana data itu berelasi, batasan ini menentukan bagaimana (harus ataukah tidak) berpartisipasi suatu entitas dengan relasinya pada entitas lain.Langkah-langkah yang dilakukan untuk menemukan atau mengidentifikasi relasi yaitu : 1) Dari gambaran cerita sistem, tandai setiap hubungan yang diwakili oleh kata kerja yang ada di dalam ilustrasi tersebut beserta entitas yang berhubungan. 2) mengidentifikasi rasio kardinalitas dari setiap hubungan. 3) mengidentifikasikan batasan partisipasi dari setiap hubungan yang ada berikut kemungkinan atribut yang muncul dari setiap hubungan. 4) Menggambarkan hubungan tersebut dalam bentuk notasi diagram dan menggabungkan dengan notasi Entitas dan atribut yang dibuat sebelumnya.

.

**d. Tugas : Mengoperasikan Aplikasi basis data**

Dalam kegiatan ini peserta didik akan melakukan eksperimen atau praktikum secara berkelompok satu kelompok terdiri dari dua sampai tiga orang. Dalam eksperimen ini peserta didik akan mengidentifikasi relasi suatu basis data dan membuat *entity relationship diagram*. Topik bisa ditentukan sendiri atau memilih beberapa alternatif seperti: basis data persewaan buku, mobil, DVD, Penjualan buku, ATK , komputer, HP, basis data kependudukan, pelatihan atau kursus, jasa perbaikan, mobil, barang elektonik dan lain-lain. Bacalah seluruh langkah eksperimen dibawah ini kemudian lakukan dengan cermat dan teliti instruksi yang diberikan:

1. Berdasarkan diskripsi role of bisnis pada tugas kegiatan belajar 3. Temukan relasi dengan menendai setiap hubungan yang diwakili oleh kata kerja yang ada di dalam ilustrasi (role of bisnis) beserta entitas yang berhubungan
2. Identifikasikan hubungan antar entitas berdasarkan langkah 1. Tampilkan hasilnya dalam tabel yang terdiri dari tiga kolom yaitu entitas-1, hubungan atau relasi dan entitas-2.
3. Identifikasikan rasio kardinalitas dari setiap hubungan antar entitas pada langkah 2. Tampilkan hasilnya dalam tabel yang terdiri dari

lima kolom yaitu : entitas-1, banyaknya entitas 1 yang berpartisipasi, hubungan atau relasi entitas-2 dan banyaknya entitas-2 yang berpartisipasi.

4. Indentifikasikan batasan partisipasi (min, max) antar entitas dari setiap hubungan pada langkah langkah 3. Tampilkan hasilnya ke dalam tabel.
5. Gambarkan entity relationship diagram secara lengkap untuk sistem basis tersebut.
6. Kumpulkan data-data setiap langkah dan analisis data tersebut menggunakan analisis diskriptif.
7. Diskusi dan komunikasikan hasilnya dalam kelompok dan buatlah kesimpulan.
8. Buatlah Laporan dan komunikasikan hasil laporan dan pembahasan dengan guru pembimbing.

**e. Test Formatif.**

Dalam test ini setiap peserta didik membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

1. Jelaskan secara singkat definisi relasi entitas dan jenis-jenis relasi basis data dan berikan contohnya ?
2. Jelaskan secara singkat definisi batasan partisipasi (constraint) dan berikan contohnya. ?
3. Jelaskan langkah-langkah untuk menemukan atau mengidentifikasi relasi ?

**f. Lembar Jawaban Test Formatif (LJ).**

**LJ- 01 : Definisi relasi, jenis-jenis relasi dan contohnya.**

.................................................................................................................
.................................................................................................................
.................................................................................................................
.................................................................................................................
.................................................................................................................

## Kegiatan belajar 5: Mapping Relasi Entitas ke Relasi Tabel

**a. Tujuan Pembelajaran.**

Setelah mengikuti kegiatan belajar 5 ini diharapkan peserta didik dapat:

- ✓ Memahami konsep mapping relasi entitas (ER) ke relasi tabel
- ✓ Menerapkan algoritma mapping relasi entitas ke relasi tabel.

**b. Uraian materi.**

**1) Algoritma Mapping Relasi Entitas (ER) Ke Relasi Tabel.**

Di dalam basis data yang menjadi pusat perhatian dan intisari sistem adalah tabel dan relasinya. Istilah tabel ini muncul dari abstraksi data pada level physical. Tabel ini sama artinya dengan entitas dari model data pada level konseptual. Setiap orang bisa membuat tabel tetapi membuat tabel yang baik tidak semua orang dapat melakukannya. Kebutuhan akan membuat tabel yang baik ini ini melahirkan beberapa teori atau metode antara lain ialah *mapping ERto table* dan Normalisasi.

Pada uraian materi ini akan menjelaskan mapping ER ke tabel sedangkan topik normalisasi akan dijelaskan dalam kegiatan 6. Algoritma atau Langkah-langkah yang dilakukan untuk memetakan ER diagram ke tabel relasional yaitu sebagai berikut:

1. Untuk setiap entitas kuat EK, buat tabel baru EK yang menyertakan seluruh simple atribut dan simple atribut dari composite atribut yang ada. Pilih salah satu atribut kunci sebagai primary key
2. Untuk setiap entitas lemah EH, buat tabel baru EH dengan mengikutsertakan seluruh simple atribut. Tambahkan primary key dari entitas kuatnya (owner entity type) yang akan digunakan sebagai primary key bersama-sama partial key dari entitas lemah
3. Untuk setiap multivalued atribut R, buatlah tabel baru R yang menyertakan atribut dari multivalue tersebut. Tambahkan primary key dari relasi yang memiliki multivalue tersebut. Kedua atribut tersebut membentuk primary key dari tabel R
4. Untuk setiap relasi binary 1:1, tambahkan primary key dari sisi yang lebih “ringan” ke sisi (entitas) yang lebih “berat”. Suatu sisi dianggap lebih

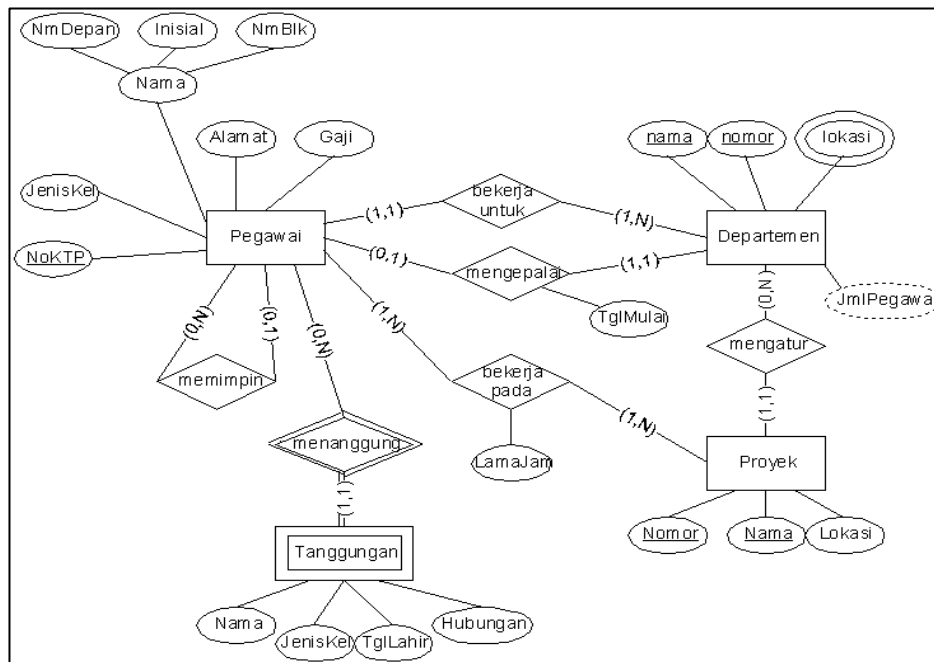
"berat" timbangannya apabila mempunyai partisipasi total. Tambahkan juga simple atribut yang terdapat pada relasi tersebut ke sisi yang lebih "berat". Apabila kedua partisipasi adalah sama-sama total atau sama-sama partial, maka dua entitas tersebut boleh digabung menjadi satu tabel

5. Untuk setiap relasi binary 1:N yang tidak melibatkan entitas lemah, tentukan mana sisi yang lebih "berat" (sisi N). Tambahkan primary key dari sisi yang "ringan" ke tabel sisi yang lebih "berat". Tambahkan juga seluruh simple atribut yang terdapat pada relasi biner tersebut
6. Untuk setiap relasi binary M:N, buatlah tabel baru R dengan atribut seluruh simple atribut yang terdapat pada relasi biner tersebut. Tambahkan primary key yang terdapat pada kedua sisi ke tabel R. Kedua foreign key yang didapat dari kedua sisi tersebut digabung menjadi satu membentuk primary key dari tabel R
7. Untuk setiap relasi lebih dari dua entitas, n-nary (ternary), meliputi dua alternatif yaitu:
    1. Buatlah tabel R yang menyertakan seluruh primary key dari entitas yang ikut serta. Sejumlah n foreign key tersebut akan membentuk primary key untuk tabel R. Tambahkan seluruh simple atribut yang terdapat pada relasi n-ary tersebut.
    2. Mengubah bentuk relasi ternary menjadi entitas lemah, kemudian memperbaiki relasi yang terjadi antara entitas lemah tersebut dengan entitas-entitas kuatnya dan melakukan algoritma mapping sesuai dengan aturan mapping.

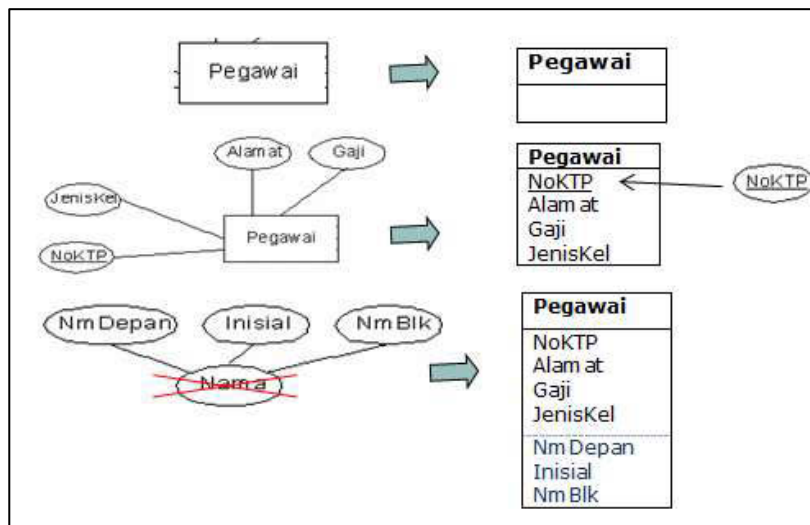
**2) Contoh Mapping ER Ke Tabel Sistem Basis Data Perusahaan.**

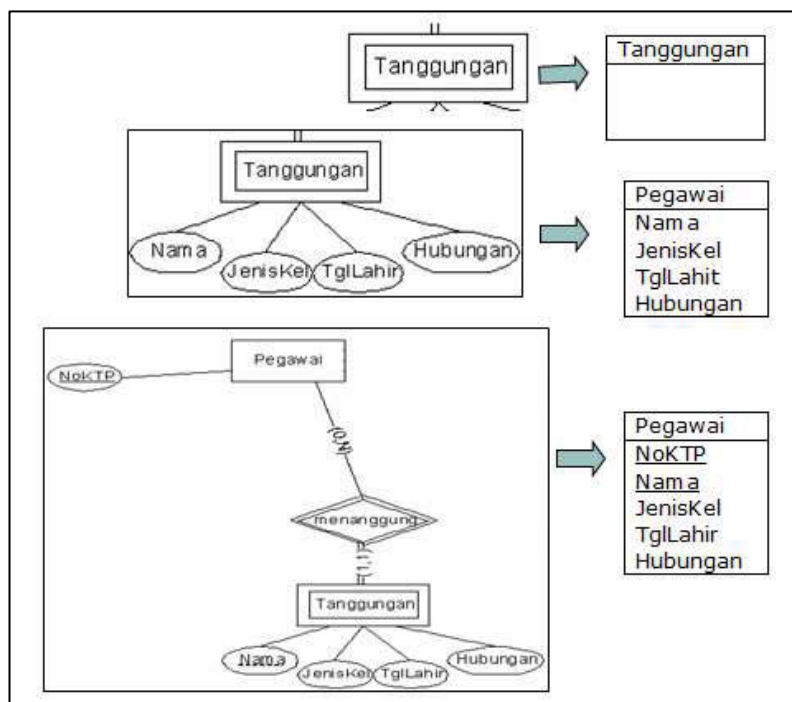
Uraian dibawah ini menjelaskan urutan langkah memetakan ER ke relasi tabel. Kasus yang diambil adalah sistem basis data perusahaan A seperti dijelaskan dalam kegiatan belajar 3 dan 4.

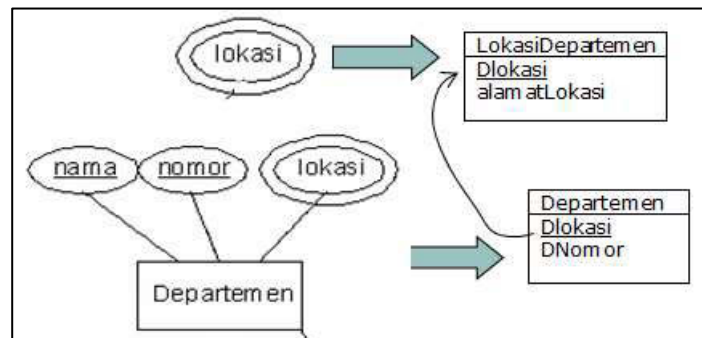
Soal :

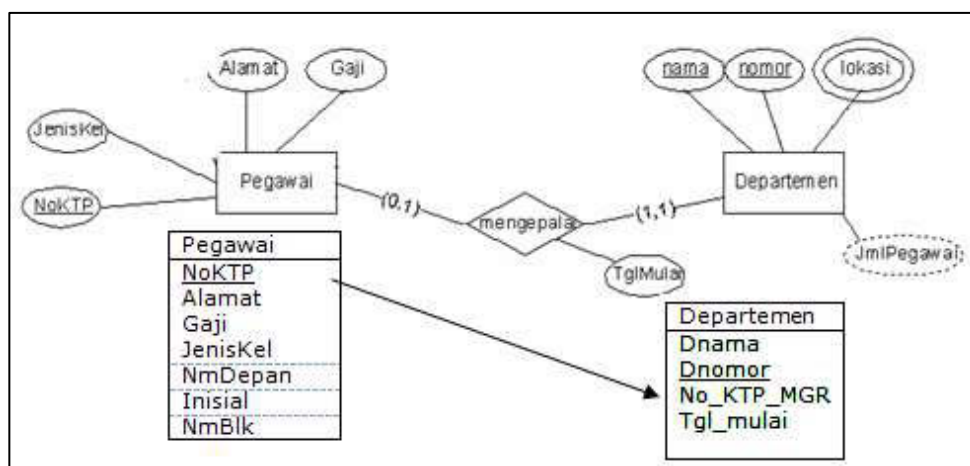
- Bacalah kembali uraian materi dan kegiatan belajar 3 dan 4.
- Berdasarkan uraian materi tentang algoritma mapping ER ke tabel buatlah relasi antar tabel dari ER diagram sistem basis data perusahaan A seperti telah dijelaskan dalam uraian materi kegiatan belajar 3 dan 4.

Gambar 19. ER Diagram sistem basis data perusahaan A

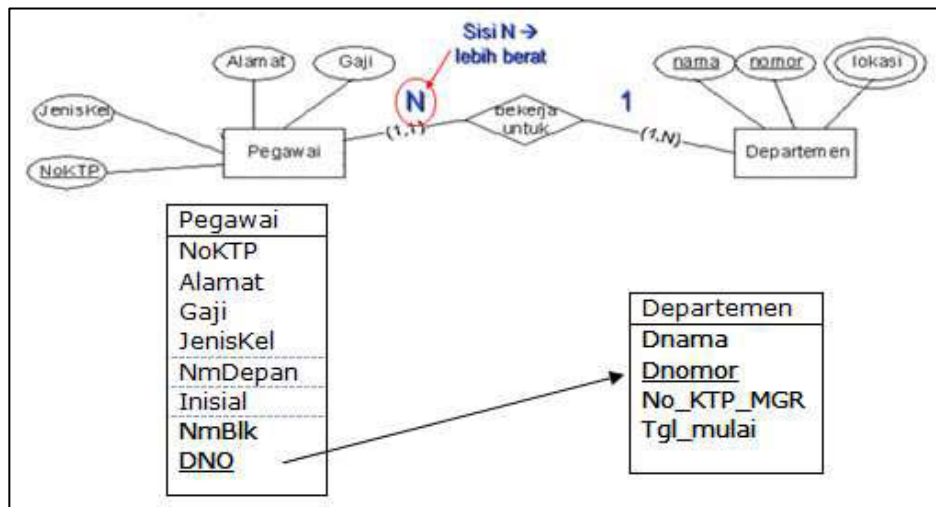
Penyelesaian :

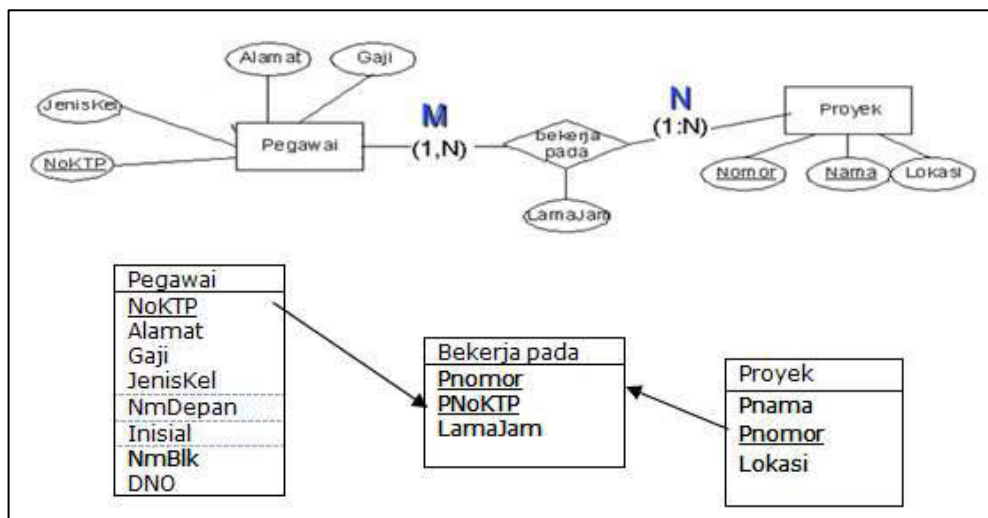
1. Berdasarkan algoritma nomor 1 aturan tentang entitas kuat maka lakukan beberapa langkah dibawah ini :
   a. Untuk setiap entitas kuat Entitas Kuat, buat tabel baru Eks.
   b. Sertakan seluruh simple atribut.
   c. Sertakan simple atribut dari composite atribut yang ada.
   d. Pilih salah satu atribut kunci sebagai primary key.

Gambar 20. Mapping ER ke tabel untuk entitas kuat

2. Berdasarkan algoritma nomor 2 aturan tentang entitas entitas lemah. Untuk setiap entitas lemah EH, laakukan beberapa langkah dibawah ini :
   a. Buat tabel baru EH.
   b. Sertakan seluruh simple atribut
   c. Tambahkan primary key dari entitas kuatnya (owner entity type) yang akan digunakan sebagai primary key bersama-sama partial key dari entitas lemah.

Gambar 21. Mapping ER ke tabel untuk entitas lemah

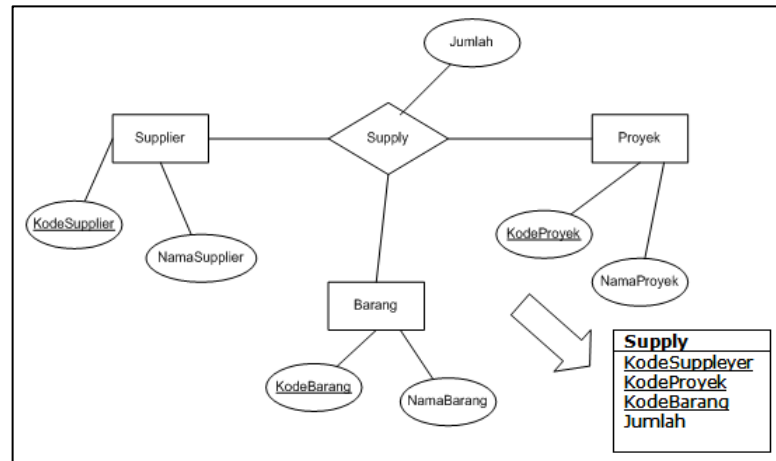
3. Berdasarkan algoritma nomor 2 aturan tentang relasi multivalue atribut.Untuk setiap multivalued atribut R,
   a. buatlah tabel baru R yang menyertakan atribut dari multivalue tersebut.
   b. Tambahkan primary key dari relasi yang memiliki multivalue tersebut. Kedua atribut tersebut membentuk primary key dari tabel R

Gambar 22. Mapping multivalue atribute

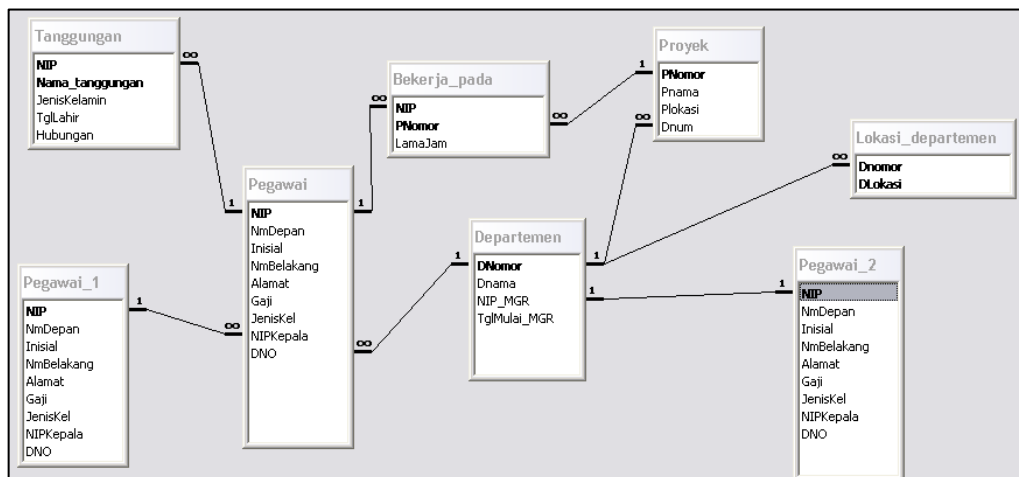
4. Untuk setiap relasi binary 1:1, tambahkan primary key dari sisi yang lebih "ringan" ke sisi (entitas) yang lebih "berat". Suatu sisi dianggap lebih "berat" timbangannya apabila mempunyai partisipasi total. Tambahkan juga simple atribut yang terdapat pada relasi tersebut ke sisi yang lebih "berat".

Gambar 23. Mapping relasi binary 1:1

5. Untuk setiap relasi binary 1:N yang tidak melibatkan entitas lemah, tentukan mana sisi yang lebih "berat" (sisi N). Tambahkan primary key dari sisi yang "ringan" ke tabel sisi yang lebih "berat". Tambahkan juga seluruh simple atribut yang terdapat pada relasi biner tersebut

Gambar 24. Mapping ER to tabel relasi *one to many*

6. Untuk setiap relasi binary M:N, buatlah tabel baru R dengan atribut seluruh simple atribut yang terdapat pada relasi biner tersebut. Tambahkan primary key yang terdapat pada kedua sisi ke tabel R. Kedua foreign key yang didapat dari kedua sisi tersebut digabung menjadi satu membentuk primary key dari tabel R

Gambar 25.Mapping ER to tabel relasi *one to many*

7. Untuk setiap relasi n-ary (ternary),
   a. Buatlah tabel R yang menyertakan seluruh primary key dari entitas yang ikut serta. Sejumlah n foreign key tersebut akan membentuk primary key untuk tabel R. Tambahkan seluruh simple atribut yang terdapat pada relasi n-ary tersebut.

b. Sama dengan proses yang dilakukan untuk langkah ke 6. Karena dalam ER-D perusahaan ini tidak ada relasi n-ary maka langkah ini tidak dilakukan.

Gambar 26. Mapping untuk relasi N-narry

Dengan menggunakan cara yang sama dapat dilakukan pemetaan ER diagram ke tabel untuk setiap relasi entitas dari ER diagram sistem basis data perusahaan A.

Gambar 27. Relasi Tabel hasil pemetakan ERD

**c. Rangkuman**

Di dalam basis data yang menjadi pusat perhatian dan intisari sistem adalah tabel dan relasinya. Istilah tabel ini muncul dari abstraksi data pada level physical. Tabel ini sama artinya dengan entitas dari model data pada level konseptual.Kebutuhan akan membuat tabel yang baik ini ini melahirkan beberapa teori atau metode antara lain ialah *mapping ERto table* dan Normalisasi.

Algoritma atau Langkah-langkah yang dilakukan untuk memetakan ER diagram ke tabel relasional meliputi tujuh aturan yaitu : 1) ketentuan entitas kuat. 2) ketentuan entitas lemah. 3) ketentuan atribut muultivalue. 4) Ketentuan relasi binary *one to one*. 5) ketentuan Ketentuan relasi binary one to many. 6) Ketentuan relasi binary many to many. 7) Ketentuan relasi t*ernary* (n-narry)

**d. Tugas : Mengoperasikan Aplikasi basis data**

Dalam kegiatan ini peserta didik akan melakukan eksperimen atau praktikum secara berkelompok satu kelompok terdiri dari dua sampai tiga orang. Dalam eksperimen ini peserta didik akan memetakan ERD yang telah dibuat pada tugas kegiatan belajar 4 ke dalam relasi tabel sehingga menjadi basis data relasional. Bacalah seluruh langkah eksperimen dibawah ini kemudian lakukan dengan cermat dan teliti instruksi yang diberikan:

1. Amatilah ERD sistem basis data yang telah dibuat dalam tugas kegiatan belajar 4.
2. Identifikasi entitas kuat, entitas lemah, multivalue atribut, relasi *one to one*, relasi *one to many,* relasi *many to many* dan relasi ternary. Tampilkan hasilnya dalam tabel.
3. Dari tabel hasil pada langkah 2 buatlah petakan ERD tersebut ke dalam relasi tabel, mulai dari menggambarkan tabel, menambahkan atribute ke dalam tabel dan menghubungkan satu tabel dengan tabel lainnya. Hasilnya dalam bentuk gambar relasi tabel.
4. Kumpulkan data-data setiap langkah dan analisis data tersebut menggunakan analisis diskriptif.
5. Diskusi dan komunikasikan hasilnya dalam kelompok dan buatlah kesimpulan.
6. Buatlah Laporan dan komunikasikan hasil laporan dan pembahasan dengan guru pembimbing

**e. Test Formatif.**

Dalam test ini setiap peserta didik membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi dan tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan**.**

1. Jelaskan secara singkat definisi tentang diagram relasi tabel dan perbedaannya diagram relasi entitas ?
2. Jelaskan algoritma atau langkah-langkah mapping ER ke tabel dan berikan contoh ?

**f. Lembar Jawaban Test Formatif (LJ).**

**LJ- 01 : Definisi relasi tabel dan perbedaanny dengan diagram relasi entitas (ERD).**

..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................
..........................................................................................................................

**LJ- 02 : Algoritma mapping ER ke tabel dan contohnya.**

g. **Lembar Kerja Peserta Didik.**

## Kegiatan belajar 6 : Model Hirarki Basis Data (Hierarchical Model)

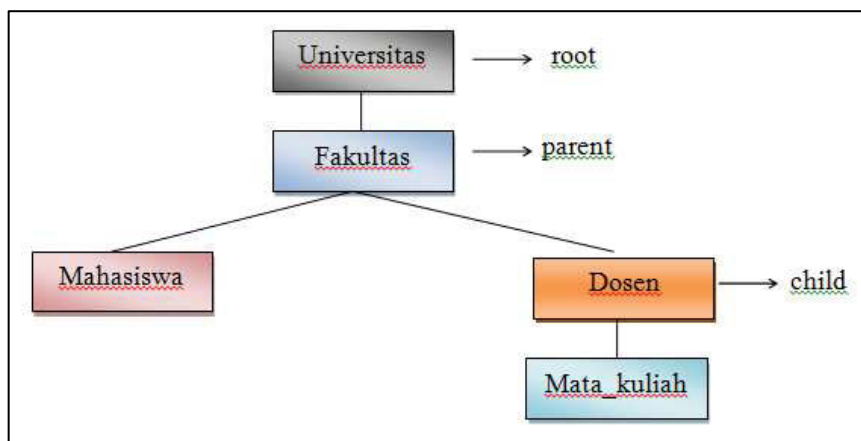
**a. Tujuan Pembelajaran.**

Setelah mengikuti kegiatan belajar 6 ini diharapkan peserta didik dapat:

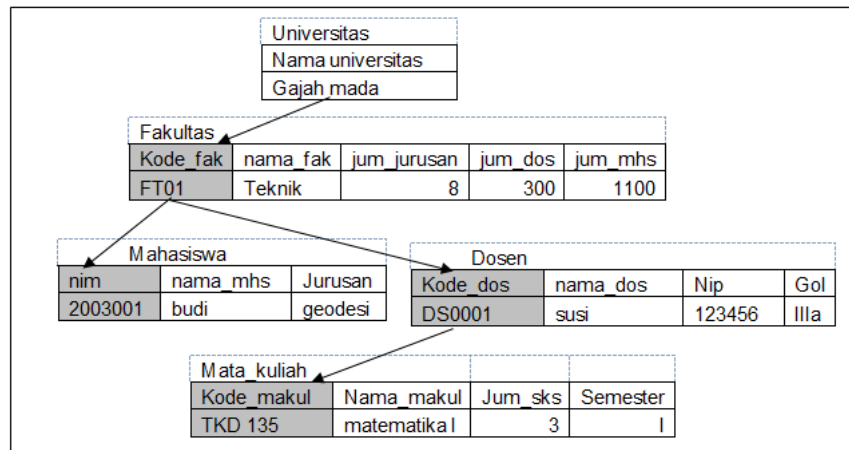
- ✓ Memahami konsep model Hirarki basis data
- ✓ Memahami jaringan basis data sebagai pengembangan model hirarki.
- ✓ Membuat struktur hirarki basis data
- ✓ Membuat struktur jaringan basis data

**b. Uraian materi.**

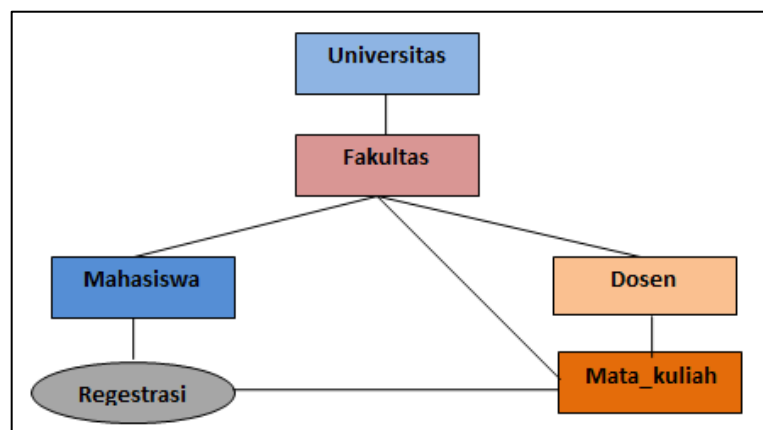
**1) Model Hirarki Basis Data (*Hierarchical Model*)**

Dalam model ini data disusun menurut struktur pohon. Puncak dari herarki disebut dengan *root* sedangkan entitas atau interface di bawahnya dikenal sebagai induk (parent). Entitas induk mempunyai beberapa sub entitsas yang disebut anak (child). Entitas dalam model hirarki dilambangkan dengan empat persegi panjang. Sedangkan relasi atau hubungan dengan entitas lain dinotasikan dengan garis. Gambar dibawah ini menjelaskan salah satu contoh model hirarki basis data level konseptual sistem perkuliahan

Gambar 28. Model Herarki sistem perkuliahan (level konseptual)

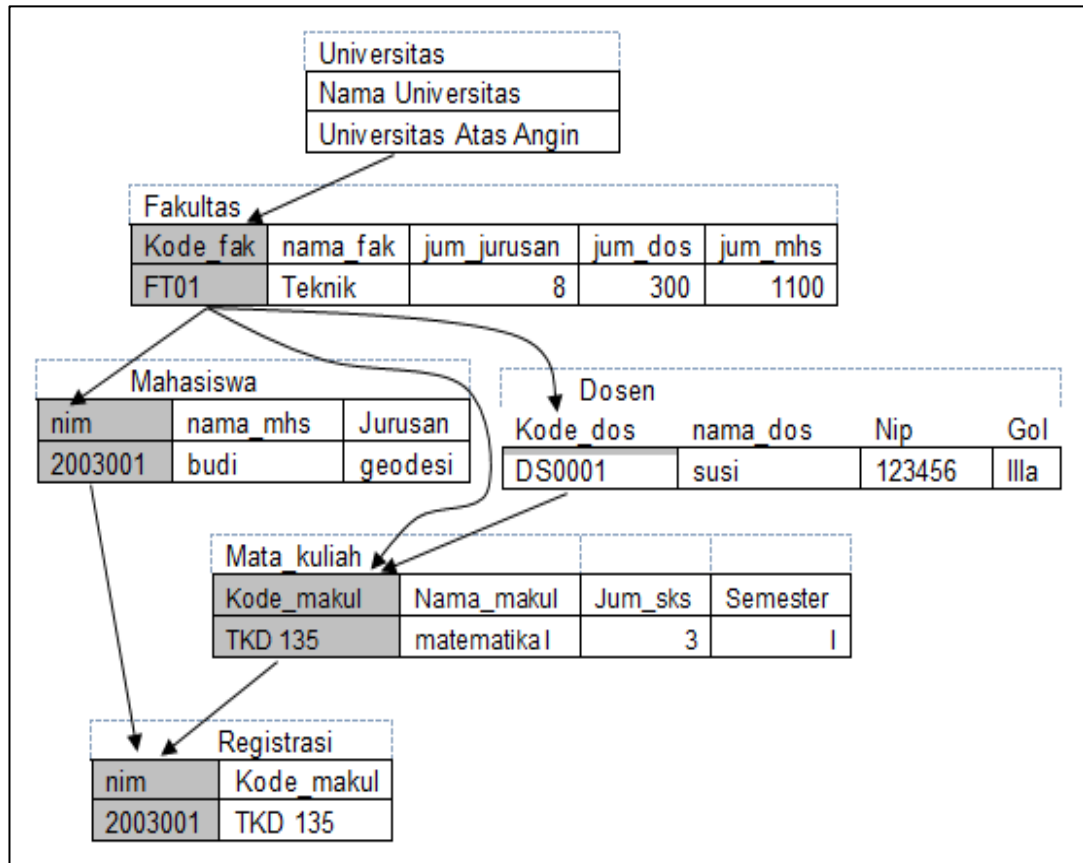
Dari gambar struktur hirarki basis data diatas dapat dibuat struktur pengkodean record data (level fisik) untuk setiap entitas beserta hubungan antar. Susuan herarkhi ditujukkan dengan tanda anak panah pada medan data (field) yang digunakan sebagai kunci data (primary key, daerah diarsir). Relasi dalam herarkhi model hubungan antar entitas dinyatakan dalam satu-banyak(one to

many) atau satu – satu (one to one). Kelemahan hararkhi model adalah tidak dapat dilakukan pencarian data pada field atribut, misal tidak dapat menampilkan data pda tabel mata kuliah berdasarkan jum_SKS, karena jum_SKS bukan kunci data. Masalah ini dapat diatasi dengan mengubah struktur data dengan memberi hubungan khusus (misalnya dengan variabel pointer).

Gambar 29. Struktur pengkodean record data (model level fisik)

**2) Model Jaringan Basis Data (Network Model).**

Dalam model jaringan entitas induk maupun anak bisa lebih dari dua. Model ini merupakan pengembangan model hirarki. Relasi antara entitas dalam network model adalah satu ke satu (*one to one*) atau satu ke banyak *(one to many)*.

Gambar 30. Model struktur jaringan basis data

Dalam network data model tidak diperbolehkan terdapat relasi banyak ke banyak (many to many). Untuk membuat relasi many to many dalam network

model dibutuhkan entitas perantara yang disebut sebagai rekaman silang (intersection record).Dari gambar 11 entitas registrasi adalah merupakan entitas perantara antara etitas mahasiswa dengan entitas mata kuliah

Gambar 31. Organisasi record data pada model jaringan

### c. Rangkuman

Model struktur hirarki basis data adalah satu model yang data disusun menurut struktur pohon. Puncak dari herarki disebut dengan *root* sedangkan entitas atau interface di bawahnya dikenal sebagai induk (parent). Entitas induk mempunyai beberapa sub entitsas yang disebut anak (child). Kelemahan hararkhi model adalah tidak dapat dilakukan pencarian data pada field atribut. Masalah ini dapat diatasi dengan mengubah struktur data dengan memberi hubungan khusus (misalnya dengan variabel pointer).

Model jaringan merupakan pengembangan model hirarki. Dalam model ini entitas induk maupun anak dapat memiliki lebih dari dua entitas. Hubungan atau relasi antara entitas dalam network model adalah satu ke satu (*one to one*) atau

satu ke banyak *(one to many)*. Ciri khas model ini terdapat adalah terdapatnya entitas perantara yang disebut sebagai rekaman silang (intersection record). Entitas perantara berfungsi untuk relasi *many to many.*

.

**d. Tugas : Mengoperasikan Aplikasi basis data**

Dalam kegiatan ini peserta didik akan melakukan eksperimen atau praktikum secara berkelompok satu kelompok terdiri dari dua sampai tiga orang.. Eksperimen dilakukan melalui pembuatan rancangan struktur hirarki dan struktur jaringan basis data. Topik bisa ditentukan sendiri atau memilih beberapa alternatif seperti: basis data persewaan buku, mobil, DVD, Penjualan buku, ATK , komputer, HP, basis data kependudukan, pelatihan atau kursus, jasa perbaikan, mobil, barang elektonik dan lain-lain. Bacalah seluruh langkah eksperimen dibawah ini kemudian lakukan dengan cermat dan teliti dengan perangkat yang telah disediakan.

1. Diskusikan dalam kelompok dan tentukan topik atau judul sistem basis data yang akan dibuat, koordinasikan dan konsultasikan dengan guru atau teknisi.
2. Buatlah diskripsi singkat tentang sistem basis data yang telah ditentukan.
3. Idetifikasikan entitas entitas yang ada termasuk entitas perantara sesuai dengan diskripsi yang telah dibuat. Buat pula diskripsi singkat setiap entitas tersebut.
4. Buatlah diagram struktur jaringan basis data sesuai dengan hasil diskripsi sistem basis data dan identifikasi entitas. Gunakan notasi-notasi yang telah distandarkan.
5. Untuk setiap entitas identifikasikan atribut yang dimiliki oleh entitas-entitas tersebut.
6. Buatlah diagram struktur pengorganisasian rekord sesuai dengan diagram struktur jaringan dan identifikasi atribut (langkah 4 dan 5). Gunakan notasi-notasi yang telah distandarkan.
7. Kumpulkan data-data setiap langkah dan analisis data tersebut menggunakan analisis diskriptif.

8. Diskusi dan komunikasikan hasilnya dalam kelompok dan buatlah kesimpulan.
11. Buatlah Laporan dan komunikasikan hasil laporan dan pembahasan dengan guru pembimbing.

**e. Test Formatif.**

Dalam test ini setiap peserta didik membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

1. Jelaskan secara singkat apa yang dimaksud dengan model struktur hirarki basis data ?
2. Jelaskan secara singkat apa yang dimaksud dengan model struktur jaringan basis data ?
3. Jelaskan perbedaan antara model diagram hirarki, model diagram jaringan dan model ERD ?.

**f. Lembar Jawaban Test Formatif (LJ).**

**LJ- 01 : Pengertian Model struktur hirarki basis data.**

.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................

g. **Lembar Kerja Peserta Didik.**

## Kegiatan belajar 7 : Ketergantungan Fungsional

**a. Tujuan Pembelajaran.**

Setelah mengikuti kegiatan belajar 7 ini diharapkan peserta didik dapat:

- ✓ Memahami konsepketergantungan fungsional.
- ✓ Menguji tabel melalui identifikasi ketergantungan fungsional.

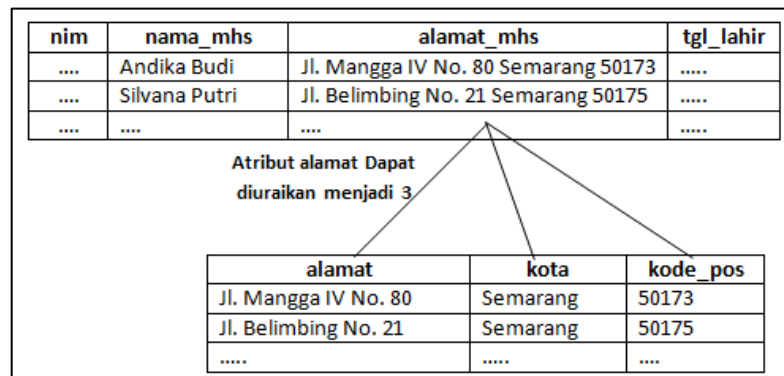
**b. Uraian materi.**

**1) Tabel dan atribut**

Sebagaimana telah dijelaskan dalam kegiatan belajar 2, tabel adalah kompulan data yang tersusun dalam format baris (record) dan kolom (field atau atribut). Atribut ini lebih umum digunakan dalam perancangan basis data, karena menunjukkan fungsinya sebagai pembentuk karakteristik (sifat–sifat) yang melekat pada sebuah table. Atribut dibedakan menjadi beberapa jenis yaitu : 1) simple atribut (*Simple Attribute*) dan Atribut Komposit (*Composite Attribute*). 2) Atribut BernilaiTunggal *(Single Valued Attribute)* dan Atribut Bernilai Banyak *(Multi Valued Atrtribute)*. 4) *Mandatory dan Non mandatory Attribute* 5) Not Null atribu 6) Atribut Turunan.

| nim | nama_mhs | alamat_mhs | tgl_lahir |
|---|---|---|---|
| .... | Andika Budi | Jl. Mangga IV No. 80 Semarang 50173 | ..... |
| .... | Silvana Putri | Jl. Belimbing No. 21 Semarang 50175 | ..... |
| .... | .... | .... | ..... |



| alamat | kota | kode_pos |
|---|---|---|
| Jl. Mangga IV No. 80 | Semarang | 50173 |
| Jl. Belimbing No. 21 | Semarang | 50175 |
| ..... | ..... | .... |

Gambar 29. Tabel data mahasiswa dengan atribut multivalue

Dari gambar diatas dan gambar dibawah ini dapat diketahui dan dikelompokkan atribut-atributnya berdasarkan kategorinya yaitu sebagai berikut :

- Atribut sederhana: nim, tgl_lahir, kota, kode pos. Merupakan atribut sederhana (atomic attribute) karena tidak bisa dipecah lagi.
- Atribut komposit : alamat_mhs(merupakan atribut komposite karena bisa dipecah lagi menjadi tabel: alamat, kota, kode pos), nama_mhs

(merupakan atribut komposite karena bisa dipecah lagi menjadi tabel: nama depan, nama belakang, inisial).

| nim | nama_mhs | alamat_mhs | tgl_lahir | hobi |
|---|---|---|---|---|
| 040001 | ..... | ..... | ..... | Membaca<br>Berenang<br>Menari |
| 040002 | ..... | ..... | ..... | |
| 040003 | ..... | ..... | ..... | Musik |
| 040004 | ..... | ..... | ..... | Sepak Bola<br>Basket |
| 040005 | ..... | ..... | ..... | Menyanyi |

Atribut Bernilai Tunggal — Atribut Bernilai Banyak

Gambar 30. Tabel mahasiswa dengan atribut tunggal dan banyak

- Atribut tunggal : nim, nama_mhs, alamat_mhs, tgl_lahir (merupakan atribut tunggal karena hanya mempunyai satu nilai)
- Atribut bernilai banyak: hobi (merupakan *multivalue atribut* karena mempunyai nilai banyak dan nilai yang jenisnya)

| nim | nama_mhs | alamat_mhs | tgl_lahir | hobi |
|---|---|---|---|---|
| ..... | ..... | ..... | ..... | ..... |
| ..... | ..... | ..... | ..... | ..... |
| ..... | ..... | ..... | ..... | ..... |
| ..... | ..... | ..... | ..... | ..... |

Mandatory Attribute — Non Mandatory Attribute

Gambar 31. Tabel mahasiswa dengan atribut mandatory

- Mandatori atribut : nim, nama-mhs (merupakan mandatori atribut karena atribut tersebut harus memiliki nilai dan tidak boleh kosong)
- Non mandatory atribut: alamat, tgl lahir, hobi (merupakan non mandatori atribute karena boleh tidak memiliki nilai atau NOT NULL)
- Atribut turunan : indeks prestasi (ip), merupakan atribut diturunkan daribeberapa atribut nilai mata kuliah.

| nim | nama_mhs | alamat_mhs | tgl_lahir | angkatan | ip |
|---|---|---|---|---|---|
| 040001 | ..... | ..... | ..... | 2004 | 3,25 |
| 040002 | ..... | ..... | ..... | 2004 | 2,85 |
| ..... | ..... | ..... | ..... | ..... | ..... |
| 050001 | ..... | ..... | ..... | 2005 | 2,95 |
| 050002 | ..... | ..... | ..... | 2005 | 3,10 |
| ..... | ..... | ..... | ..... | ..... | ..... |

Atribut Turunan

Gambar 32. Tabel mahasiswa dengan atribut turunan.

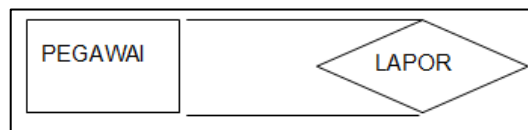
2) **Relationship**

Sebagai mana dijelaskan pada uraian kegiatan 3 bahwa relasi atau *relationship* merupakan hubungan yang terjadi antara satu atau lebih entitas. Berikut ini adalah contoh penggambaran diagram relationship antara 2 entitas :

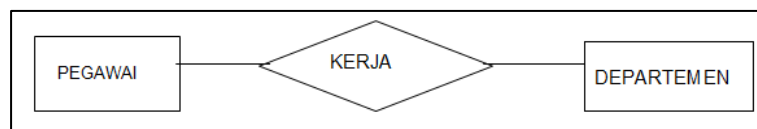
Gambar 33. Contoh diagram relationship antar 2 entitas

Pada diagram di atas terlihat relasi ‘kerja’ antara entitas pegawai dengan entitas proyek.Derajat dari relationship menjelaskan jumlah entitas yang berpartisipasi dalam suatu relationship. Klasifikasi relasi berdasarkan derajatnya adalah:

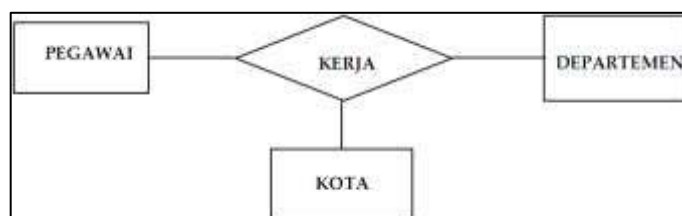
1. Unary Degree (Derajat Satu)

   Relationship yang hanya melibatkan 1 entitas.

*Gambar 34. Unary Degree Relationship*

2. Binary Degree (Derajat Dua)

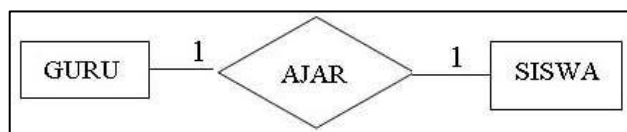
   Relationship yang melibatkan 2 entitas.

Gambar 35. Binary Degree Relationship Relationship

3. Ternary Degree (Derajat tiga)

   Relationship yang melibatkan 3 entitas.

Gambar 36 Ternary Degree Relationship Relationship

*Rasio kardinalitas*atau Cardinality Ratio Constraintmenjelaskan batasan jumlah keterhubungan satu entitas dengan entitas lainnya.  Jeni – jenis *Cardinality Ratio*(rasio kardinalitas)

1. One–to–one (1 : 1)

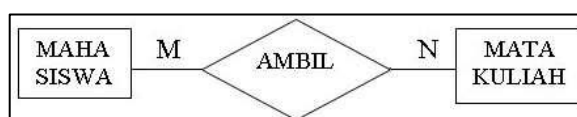
   Relationship antar entitas dimana hubungan antara entitas pertama dan kedua adalah satu berbanding satu.Contoh : pada pengajaran private satu guru satu siswa. “seorang guru mengajar seorang siswa, seorang siswa diajar oleh seorang guru”

Gambar 37. Kardinalitas one to one

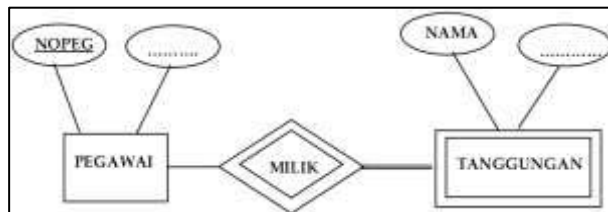
2. One-to-manyatau many-to-one(1 : N atau N : 1)

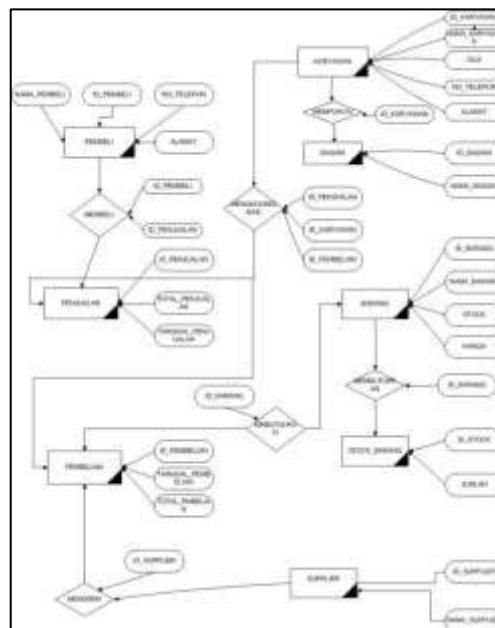
   Relationship antar entitas dimana hubungan antara entitas pertama dan kedua adalah satu berbanding banyak atau banyak berbanding satu.Contoh kasus :Dalam suatu perusahan satu bagian mempekerjakan banyak pegawai. “Satu bagian mempekerjakan banyak pegawai, satu pegawai kerja dalam satu bagian”

Gambar 38  Kardinalitas one to many

3. Many-to-many (N : N)

   Relationship antar entitas dimana hubungan antara entitas pertama dan kedua adalah banyak berbanding banyak.Contoh kasus : Dalam Dalam universitas seorang mahasiswa dapat mengambil banyak matakuliah.“Satu mahasiswa mengambil banyak matakulih dan satu matakuliah diambil banyak mahasiswa.”

Gambar 39  Kardinalitas many to many

Weak Entity adalah suatu entitas dimana keberadaan dari entitas tersebut tergantung dari keberadaan entitas lain. Entitas yang merupakan induknya disebut *Identifying Owner* dan relationshipnya disebut *Identifying Relationship*. Contoh dari sebuah weak entity :

Gambar 40. Weak entity

Entitas tanggungan adalah weak entity karena tergantung penuh pada entitas pegawai. Berikut ini adalah contoh ER-Diagram dari sebuah database toko :

Gambar 41. Contoh ER-Diagram dari sebuah database toko

### 3) Ketergantungan Fungsional

Atribut Y pada relasi R dikatakan tergantung fungsional pada atribut X (R,X → R,Y), jika dan hanya jika setiap nilai X pada relasi R mempunyai tepat satu nilai Y pada R.Misal, terdapat skema database Pemasok_barang. Dengan tabel pemasok (No_pem, Na_pem)

| No_pem | Na_pem |
|---|---|
| P01 | Baharu |
| P02 | Sinar |
| P03 | Harapan |

Ketergantungan fungsional dari tabel Pemasok_barang adalah :

No_pem → Na_pem (nama pemasuk tergantung secara fungsional dari Nomer pemasok)

**4) Ketergantungan Fungsional Penuh**

Atribut Y pada relasi R dikatakan tergantung fungsional penuh pada atribut X pada relasi R, jika Y tidak tergantung pada subset dari X (bila X adalah key gabungan). Contoh tabel Kirim_barang (No_pem, Na_pem, No_bar, Jumlah)

| No_pem | Na_pem | No_bar | Jumlah |
|---|---|---|---|
| P01 | Baharu | B01 | 1000 |
| P01 | Baharu | B02 | 1500 |
| P01 | Baharu | B03 | 2000 |
| P02 | Sinar | B03 | 1000 |
| P03 | Harapan | B02 | 2000 |

Ketergantungan fungsionalnya adalah :

No_pem → Na_pem

No_bar, No_pem → Jumlah (tergantung penuh terhadap keynya)

**5) Ketergantungan Transitif**

Atribut Z pada relasi R dikatakan tergantung transitif pada atribut X, jika atribut Y tergantung pada atribut X pada relasi R dan atribut tergantung pada atribut Y pada relasi R. Contoh perhatikan tabel dibawah ini :

| No_pem | Kode_kota | Kota | No_bar | Jumlah |
|---|---|---|---|---|
| P01 | 1 | Jakarta | B01 | 1000 |
| P01 | 1 | Jakarta | B02 | 1500 |
| P01 | 1 | Jakarta | B03 | 2000 |
| P02 | 3 | Bandung | B03 | 1000 |
| P03 | 2 | Surabaya | B02 | 2000 |

Ketergantungan fungsional :

No_pem → Kode_kota

Kode_kota → Kota, maka

No_pem → Kota

**6) Contoh Lain Ketergantungan Fungsioanl.**

Diberikan sebuah tabel T berisi paling sedikit 2 buah atribut, yaitu A dan B. Kita dapat menyatakan notasi berikut ini :

A → B

Yang berarti A secara fungsional menentukan B atau B secara fungsional tergantung pada A, jika dan hanya jika setiap kumpulan baris (*row*) yang ada di tabel T, pasti ada 2 baris data (*row*) di tabel dengan nilai A yang sama, maka nilai B pasti juga sama. Definisi yang paling formal untuk itu adalah :

Diberikan 2 *row* r1 dan r2 dalam tabel T dimana A → B.

jika r1(A) = r2(A) maka r1(B) = r2(B)

| | nama_kul | nim | nama_mhs | indeks_nilai |
|---|---|---|---|---|
| *row 1* | Sistem Basis Data | 040001 | Santi Purnamasari | A |
| *row 2* | Sistem Basis Data | 040002 | Budi Setyawan | B |
| *row 3* | Struktur Data | 040001 | Santi Purnamasari | |
| *row 4* | Struktur Data | 040002 | Budi Setyawan | |
| *row 5* | Struktur Data | 040003 | Kartika Sari | |
| *row 6* | Komunikasi Data | 040001 | Santi Purnamasari | B |
| *row 7* | Riset Operasi | 040002 | Budi Setyawan | C |

Dengan melihat data di atas dan dengan pertimbangan intuisi kita, maka ketergantungan fungsional yang dapat kita ajukan adalah :

- nim → nama_mhs

  yang berarti bahwa atribut *nama_mhs* hanya tergantung pada atribut *nim*. Hal ini dibuktikan dari fakta : untuk setiap nilai *nim* yang sama maka pasti nilai *nama_mhs*nya juga sama.
- nama_kul, nim → indeks_nilai

  yang berarti bahwa atribut *indeks_nilai* tergantung pada atribut *nama_kul* dan *nim* secara bersama–sama, memang kita tidak dapat menunjukkan fakta, bahwa untuk setiap nilai *nama_kul* dan *nim* yang sama, maka nilai *indeks_nilai*nya juga sama, karena *nama_kul*, *nim* merupakan *key* (sehingga bersifat unik) untuk tabel tersebut. Tetapi, ketergantungan fungsional tersebut sesuai dengan pengertian bahwa setiap *indeks_nilai* diperuntukkan pada mahasiswa tertentu untuk mata kuliah tertentu yang diambilnya.

Tanpa memperhatikan pengertian ketergantungan secara alamiah terhadap tabel tersebut, kita juga dapat mengajukan sejumlah ketidaktergantungan (*non KF*) dengan hanya melihat fakta yang ada,yaitu :

- nama_kul $\not\rightarrow$ nim

  yang artinya atribut *nim* tidak tergantung pada atribut *nama_kul*. Buktinya terlihat pada *row* 1 dan *row* 2 : dengan nilai *nama_kul* yang sama, tapi nilai *nim*nya berbeda.
- nim $\not\rightarrow$ nideks_nilai

  yang artinya atribut *indeks_nilai* tidak bergantung pada atribut *nim*. Buktinya terlihat pada *row* 1 dan *row* 3 : dengan nilai *nim* yang sama, tapi nilai *indeks_nilai* berbeda.

**c. Rangkuman**

Tabel adalah kompulan data yang tersusun dalam format baris (record) dan kolom (field atau atribut). Atribut ini lebih umum digunakan dalam perancangan basis data, karena menunjukkan fungsinya sebagai pembentuk karakteristik (sifat–sifat) yang melekat pada sebuah tabel.

Relasi atau relationship merupakan hubungan yang terjadi antara satu atau lebih entitas. Derajat dari relationship menjelaskan jumlah entitas yang berpartisipasi dalam suatu relationship. *Rasio kardinalitas*atau Cardinality Ratio

Constraintmenjelaskan batasan jumlah keterhubungan satu entitas dengan entitas lainnya

Suatu atribut Y pada relasi R dikatakan tergantung fungsional pada atribut X (R,X → R,Y), jika dan hanya jika setiap nilai X pada relasi R mempunyai tepat satu nilai Y pada R. Suatu atribut Y pada relasi R dikatakan tergantung fungsional penuh pada atribut X pada relasi R, jika Y tidak tergantung pada subset dari X (bila X adalah key gabungan). Suatu atribut Z pada relasi R dikatakan tergantung transitif pada atribut X, jika atribut Y tergantung pada atribut X pada relasi R dan atribut tergantung pada atribut Y pada relasi R.

.

**d. Tugas : Mengamati ketergantungan fungsional**

Dalam kegiatan ini peserta didik akan melakukan eksperimen atau praktikum secara berkelompok, satu kelompok terdiri dari dua sampai tiga orang. Eksperimen dilakukan melalui pengamatan terhadap *entity relationship diagram* yang telah dibuat dalamtugas kegiatan belajar 5. Peserta didik akan mengidentifikasi ketergantungan fungsional dengan memberikan sejumlah data pada setiap tabel dalam ERD. Bacalah seluruh langkah eksperimen dibawah ini kemudian lakukan dengan cermat dan teliti dengan perangkat yang telah disediakan.

1. Lihat dan amati kembali hasil tugas belajar kegiatan 5 tentang pemetaan ER diagram ke relasi tabel.
2. Pastikan dalam relasi tabel diatas terdapat relasi one to one, relasi one to many, relasi many to many dan relasi dan relasi ternary. Jika belum ada salah satu jenis relasi tersebut buat relasi tabelnya.
3. Untuk setiap tabel dalam langkah dua di atas lengkapi data dengan menambahkan *record-record* dengan jumlah record 3 -5 record. Tampilkan hasilnya dalam tabel.
4. Dari hasil tabel pada langkah tiga di atas identifikasikan ketergantungan fungsional untuk setiap tabel. Tampilkan hasilnya dengan menggunakan tabel.
5. Kumpulkan data-data setiap langkah dan analisis data tersebut menggunakan analisis diskriptif.

6. Diskusi dan komunikasikan hasilnya dalam kelompok dan buatlah kesimpulan.
7. Buatlah Laporan dan komunikasikan hasil laporan dan pembahasan dengan guru pembimbing

**e. Test Formatif.**

Dalam test ini setiap peserta didik membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

1. Jelaskan secara singkat definisi ketergantungan fungsional dan berikan contoh ?
2. Jelaskan secara singkat definisi ketergantungan fungsional penuh dan berikan contoh ?
3. Jelaskan secara singkat definisi ketergantungan fungsional transitif dan berikan contoh ?
4. Jelaskan fungsi atau manfaat ketergantungan fungsional dalam sistem basis data?

**f. Lembar Jawaban Test Formatif (LJ).**

**LJ- 01 : Pengertian ketergantungan fungsional dan contohnya.**

..............................................................................................................

..............................................................................................................

..............................................................................................................

..............................................................................................................

..............................................................................................................

..............................................................................................................

..............................................................................................................

..............................................................................................................

..............................................................................................................

..............................................................................................................

..............................................................................................................

..............................................................................................................

..............................................................................................................

**Kegiatan belajar 8: Pengantar Teknik Normalisasi Data.**

**a. Tujuan Pembelajaran.**

Setelah mengikuti kegiatan belajar 1 ini diharapkan peserta didik dapat:

- ✓ Memahami konsepteknik normalisasi data
- ✓ Memahami tiga konsep dasar yang dibutuhkan dalam normalisasi data
- ✓ Memahami persyaratan teknik normalisasi data.
- ✓ Menguji tabel relasional menggunakan dua kriteria yaitu: *Lossless–Join Decomposition dan Dependency Preservation*

**b. Uraian materi.**

**1) Perancangan Basis Data**

Prancang basis data merupakan suatu hal yang sangat penting. Kesulitan utama dalam merancang database adalah bagaimana merancang sehingga database dapat memuaskan keperluan saat ini dan masa mendatang.Tujuan perancanganadalah agar dapat memiliki basis data yang kompak, efisien dalam penggunaan ruang penyimpanan, cepat dalam pengaksesan dan mudah dalam memanipulasi data (operasi tambah, ubah, hapus).

Perancangan model konseptual perlu dilakukan di samping perancangan model phisik.Perancangan konseptual akan menunjukkan entity dan relasinya berdasarkan proses yang diinginkan oleh organisasi. Tugas pernacangan model konseptual basis data ini merupakan tanggung jawab dari Database Administrator. Beberapa pengertian berkaiatan dengan perancangan model konseptual, ialah :

- Bukan merupakan pendekatan proses informasi untuk seorang programmer aplikasi, tetapi merupakan kombinasi beberapa cara untuk memproses data untuk beberapa aplikasi.
- Tidak tergantung pada aplikasi individual.
- Tidak tergantung pada DBMS yang digunakan.
- Tidak tergantung pada hardware yang digunakan.
- Tidak tergantung pada phisikal model.
- Tidaklah perlu dipikirkan tentang terapan dan operasi yang akan dilakukan pada database.

Pada perancangan model konseptual penekanan tinjauan dilakukan pada struktur data dan relasi antara file.Pendekatan yang dilakukan pada menggunakan model data relational.Dalam merancang basis data dapat dilakukan melalui dua pendekatan yaitu:

1. Model Entity–Relationship-diagram (telah dijelaskan dalam uraian kegiatan 3, 4 dan 5)
2. Menerapkan normalisasi terhadap struktur tabel yang telah diketahui.

**2) Definisi Normalisasi**

Normalisasi diartikan sebagai suatu teknik yang menstrukturkan atau mendekomposisi atau memecah data menggunakan cara–cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam basis data. Permasalahan yang dimaksud adalah berkaitan dengan penyimpangan–penyimpangan (*anomalies*) yang terjadi akibat adanya kerangkapan data dalam relasi dan inefisiensi pengolahan.Proses normalisasi akan menghasilkan relasi yang optimal, yaitu :

1. Memiliki struktur *record* yang mudah untuk dimengerti.
2. Memiliki struktur *record* yang sederhana dalam pemeliharaan.
3. Memiliki struktur *record* yang mudah untuk ditampilkan kembali untuk memenuhi kebutuhan pemakai.
4. Minimalisasi kerangkapan data guna meningkatkan kinerja sistem.

Dalam pendekatan normalisasi, perancangan basis data bertitik tolak dari situasi nyata. Ia telah memiliki item–item data yang siap ditempatkan dalam baris dan kolom pada tabel–tabel relasional. Demikian juga dengan sejumlah aturan tentang keterhubungan antara item–item data tersebut. Sementara pendekatan model data ER lebih tepat dilakukan jika yang diketahui baru prinsip sistem secara keseluruhan.

Pada penerapannya dua pendekatan tersebut dilakukan secara bersama–sama dan, berganti–ganti. Untuk kepentingan evaluasi dan dokumentasi, hasil normalisasi diwujudkan dalam sebuah model data. Model data yang sudah jadi tersebut bisa saja dimodifikasi dengan pertimbangan tertentu. Selanjutnya Hasil modifikasinya diimplementasikan dalam bentuk sejumlah struktur tabel dalam sebuah basis data. Struktur ini dapat diuji kembali dengan menerapkan aturan–aturan normalisasi, hingga akhirnya diperoleh sebuah struktur basis data yang

benar–benar efektif dan efisien. Begitulah kedua pendekatan dapat saling memperkuat satu sama lain.

**3) Domain dan Tipe Data Konsep pendukung teknik normalisasi.**

Beberapa konsep yang harus dipahami sebelum mengimplementasikan teknik normalisasi data antara lain ialah: 1) ketergantungan fungsional (sudah dibahas dalam kegiatan belajar tujuh). 2) Domain dan tipe data. 3) Konsep key atribut (Field/atribute kunci).

Penetapan tipe data pada setiap atribut (kolom) digunakan untuk keperluan penentuan struktur setiap tabel. Penetapan tipe data ini akan berimplikasi pada adanya batas–batas nilai yang mungkin disimpan atau diisikan kesetiap atribut tersebut. Jika telah menetapkan bahwa tipe data untuk sebuah atribut adalah integer, maka kita hanya mungkin untuk menyimpan data angka yang bulat diantara –32.768 hingga 32.768. Pengguna tidak mungkin untuk memasukkan data diluar batas nilai tersebut. untuk memasukkan data pecahan pengguna harus menggunakan data nbertipe string atau text.

Domain memiliki banyak kesamaan pengertiannya dengan fungsi tipe data tersebut. Akan tetapi, tipe data merujuk pada kemampuan penyimpanan data yang mungkin bagi suatu atribut secara fisik, tanpa melihat layak tidaknya data tersebut bila dilihat dari kenyataannya pemakaiannya. Sementara domain nilai lebih ditetapkan pada batas–batas nilai yang diperbolehkan bagi suatu atribut, dilihat dari kenyataanya yang ada.

Contoh : pada tabel kuliah, ditetapkan tipe data untuk atribut sks adalah integer. Dengan begitu secara fisik kita dapat menyimpan nilai –1, 0 atau 100 untuk atribut sks. Tetapi kita mengetahui dengan pasti, bahwa nilai–nilai tersebut tidak pantas (*invalid*) untuk menjadi data pada atribut sks. Lalu nilai–nilai yang boleh (*valid*) untuk atribut sks adalah 1, 2, 3, 4 dan 6, maka dapat dikatakan, domain nilai untuk atribut sks adalah 1, 2, 3, 4 dan 6.

**4) Key Attribute**

Setiap file atau tabel selalu mempunyai kunci (key) yaitu berupa satu field atau satu set field yang dapat mewakili record. Misalnya nomor pegawai merupakan kunci dari tabel pegawai suatu perusahaan. setiap pencarian cukup dengan menyebut nomor pegawai tersebut maka dapat diketahui nama, alamat

dan atribute lainnya mengenai seorang pegawai tersebut.*Key* adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data (row) dalam table secara unik.Macam-macam Filed/attribute kunci:

- *Candidate Key* (Kunci Kandidat/Kunci Calon)
- *Primary Key* (Kunci Primer)
- *Alternate Key* (Kunci Alternatif)
- *Foreign Key* (Kunci Tamu)

*Candidate Key* (Kunci Kandidat/Kunci calon)adalah satu atribute atau satu set minimal atribute yang mengidentifikasikan secara unik suatu kejadian spesific dari entity. Jika satu kunci kandidat berisi lebih dari satu atribute, maka biasanya disebut sebagai *composite key (kunci campuran/ gabungan).*

Misalnya tabel Pegawai berisi attribute:No Induk Pegawai (NIP) , No KTP, Nama, Tempat Lahir, Tanggal Lahir, Alamat, KotaKunci kandidat disini antara lain ialah :

- No Induk Pegawai (NIP), karena unik tidak mungkin ganda.
- No KTP, karena unik tidak mungkin ganda.
- Nama, sering dipakai sebagai kunci pencarian namun tidak dapat dikatakan kunci karena sering seseorang punya nama yang sama.
- Nama + Tanggal lahir, mungkin dapat dipakai sebagai kunci karena kemungkinan sangat kecil seseorang punya nama sama yang lahir pada hari yang sama.
- Nama + tempat lahir + tanggal lahir, dapat dipakai sebagai kunci
- Alamat, kota (bukan kunci).

*Primary Key* adalah satu atribute atau satu set minimal atribute yang tidak hanya mengidentitikasi secara unik suatu kejadian spesific, tapi juga dapat mewakili setiap kejadian dari suatu entity. Setiap *kunci kandidat* punya peluang menjadi *primary key*, tetapi sebaiknya dipilih satu saja yang dapat mewakili secara menyeluruh terhadap entity yang ada. Contoh:

- No Induk (NIP), karena unik tidak mungkin ganda dan mewakili secara menyeluruh terhadap entity Pegawai, dan setiap pegawai selalu punya nomor induk
- No KTP, ini hanya dipakai bila sampai dengan pembayaran gaji pegawai ternyata nomor induk belum keluar.

| NIP | Nama_Pegawai | Tgl_Lahir | Alamat |
|---|---|---|---|
| 2001 | Anton | 12-12-76 | Solo |
| 2002 | Budi | 02-02-75 | Yogya |
| 2003 | Anton | 11-11-76 | Semarang |

*Alternate Key* (Kunci alternatif)adalah kunci kandidat yang tidak dipakai sebagai primary key. Kerap kali kunci alternatif dipakai sebagai kunci pengurutan dalam laporan.Contoh: Kunci Alternatif untuk pengurutan berdasarkan nama.

Kunci alternatif

| NIP | Nama_Pegawai | Tgl_Lahir | Alamat |
|---|---|---|---|
| 2001 | Anton | 12-12-76 | Solo |
| 2002 | Budi | 02-02-75 | Yogya |
| 2003 | Anton | 11-11-76 | Semarang |

*Foreign Key* (Kunci Tamu/Asing)*Foreign Key* adalah satu atribute (atau satu set atribute) yang melengkapi satu relationship (hubungan) yang menunjukkan ke induknya. Kunci tamu ditempatkan pada entity anak dan sama dengan kunci primary induk direlasikan. Contoh:

Tabel Pegawai

| NIP | Nama_Pegawai | Tgl_Lahir | Alamat |
|---|---|---|---|
| 2001 | Anton | 12-12-76 | Solo |
| 2002 | Budi | 02-02-75 | Yogya |
| 2003 | Anton | 11-11-76 | Semarang |

Tabel Golongan

| Golongan | Gaji_Pokok | Tunjangan |
|---|---|---|
| 1 | 1000000 | 100000 |
| 2 | 2000000 | 200000 |
| 3 | 3000000 | 300000 |

Tabel GAJI

| No_Slip | NIP | Golongan | Gaji_Kotor | Pajak | Gaji_Bersih |
|---|---|---|---|---|---|
| S001 | 2001 | 1 | 1100000 | 5% | 1045000 |
| S002 | 2002 | 2 | 2200000 | 10% | 1980000 |
| S003 | 2003 | 2 | 2200000 | 10% | 1980000 |

***Primary Key***: NIP (table Pegawai), Golongan (Tabel Golongan), NoSLIP (Tab. Gaji)

***Foreign Key***: NIP, Golongan (pada table GAJI)

Dalam hal hubungan dua buah file yang punya relationship banyak lawan banyak maka terdapat 2 buah kunci tamu pada file konektornya. Contoh:

File Proyek berisi atribute: NomorProyek, Tgl Mulai, Tgl Selesai, Anggaran

File Pegawai berisi atribute:   No Induk, Nama

Hubungan antara file tersebut adalah banyak lawan banyak yaitu satu Pegawai mengerjakan lebih dari 1 proyek dan satu proyek dikerjakan oleh beberapa

pegawai maka untuk menunjukkan hubungan tersebut dipakai file konektor yang berisi kunci tamu dari kedua file.

File Proyek_Pegawai berisi atribute : Nomor Proyek, NIP, Jam Kerja (proyek tersebut dikerjakan oleh pegawai ter-tentu selama sekian jam kerja)

Maka pada file Proyek_Pegawai terdapat kunci tamu yaitu Nomor Proyek dan NIP. Kedua atribute tersebut juga merupakan kunci primary.

**5) Persyaratan teknik normalisasi data.**

Dalam perspektif normalisasi, sebuah basis data dapat dikatakan baik, jika setiap tabel yang menjadi unsur pembentuk basis data tersebut juga telah berada dalam keadaan baik atau normal. Selanjutnya, sebuah tabel dapat dikategorikan baik (*efisien*) atau normal, jika telah memenuhi 3 (tiga) kriteria berikut :

1. Jika ada *dekomposisi* (penguraian) tabel, maka dekomposisinya harus dijamin aman (*Lossless–Join Decomposition*).
2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (*DependencyPreservation*).
3. Tidak melanggar *Boyce–Code Normal Form* (BCNF)

**6) Lossless–Join Decomposition**

Dekomposisi memang merupakan upaya untuk mendapatkan tabel yang baik. Tetapi jika tidak hati–hati, upaya ini justru dapat menghasilkan kesalahan. Dekomposisi yang benar terjadi jika tabel–tabel hasil dekomposisi kita gabungkan kembali dapat menghasilkan tabel awal sebelum didekomposisi. Dekomposisi yang benar semacam ini disebut *Lossless–Join Decomposition* atau *Lossless Decomposition* (dekomposisi aman).

Di bawah ini contoh abstrak yang menghasilkan dekomposisi tidak aman (*Lossy–Join Decomposition*) :

Diasumsikan tabel XYZ yang didefinisikan oleh dua buah ketergantungan fungsional X → Y dan Y → Z. Kedua ketergantungan fungsional tersebut diperoleh dari pengamatan terhadap data yang kurang memadai atau karena asumsi yang kurang tepat. Katakan isi tabel XYZ tersebut adalah sebagai berikut:

| | X | Y | Z |
|---|---|---|---|
| *row 1* | x1 | 10 | z1 |
| *row 2* | x2 | 20 | z2 |
| *row 3* | x3 | 30 | z3 |
| *row 4* | x4 | 20 | z4 |

Memang dengan isi seperti itu, pernyataan ketergantungan fungsional yang kedua Y → Z tidak sepenuhnya tepat, karena pada *row* 2 dan *row* 4, dengan nilai untuk atribut Z–nya berbeda. Tetapi yang ingin ditekankan di sini adalah adanya dua buah ketergantunagn fungsional itu mendorong kita untuk mendekomposisi tabel XYZ tersebut menjadi dua buah tabel yaitu tabel XY dan YZ sebagai berikut :

| X | Y |
|---|---|
| x1 | 10 |
| x2 | 20 |
| x3 | 30 |
| x4 | 20 |

| Y | Z |
|---|---|
| 10 | z1 |
| 20 | z2 |
| 30 | z3 |
| 20 | z4 |

Jika kedua tabel diatas kita gabungkan kembali, maka hasilnya adalah :

| X | Y | Z |
|---|---|---|
| x1 | 10 | z1 |
| x2 | 20 | z2 |
| x2 | 20 | z4 |
| x3 | 30 | z3 |
| x4 | 20 | z2 |
| x4 | 20 | z4 |

Maka ini tentu saja berbeda dengan tabel awal (sebelum didekomposisi). Maka dekomposisi semacam ini disebut *Lossy–Join Decomposition* (dekomposisi tidak aman), yaitu sebuah dekomposisi yang sedapat mungkin kita hindari.Akan tetapi jika data pada *row 4* yang ada di tabel XYZ awal, kita ganti dengan data berikut :

| x4 | 20 | z2 |
|---|---|---|

Sehingga tabel XYZ menjadi :

| | X | Y | Z |
|---|---|---|---|
| *Row 1* | x1 | 10 | z1 |
| *Row 2* | x2 | 20 | z2 |

| | | | |
|---|---|---|---|
| *Row 3* | x3 | 30 | z3 |
| *Row 4* | x4 | 20 | z2 |

Dengan data ditabel XYZ demikian maka kedua ketergantungan fungsional dapat dibenarkan. Tabel XYZ tersebut didekomposisikan menjadi tabel XY dan YZ sebagai berikut :

| X | Y |
|---|---|
| x1 | 10 |
| x2 | 20 |
| x3 | 30 |
| x4 | 20 |

| Y | Z |
|---|---|
| 10 | z1 |
| 20 | z2 |
| 30 | z3 |

Kalau kedua tabel diatas digabungkan kembali maka, tabel awal XYZnya akan diperoleh kembali sehingga dekomposisi tersebut aman. Karena itulah ketergantungan fungsional pada suatu tabel harus kita tetapkan berdasarkan pengamatan yang teliti dan asumsi yang dapat dipertanggung jawabkan agar kelak hasil dekomposisi dapat dibenarkan.

**7) *Dependency Preservation***

*Dependency Preservation* (pemeliharaan ketergantungan) merupakan kriteria kedua yang harus dapat dicapai untuk mendapatkan tabel dan basis data yang baik. Ketika kita melakukan perubahan data, maka harus bisa dijamin agar perubahan tersebut tidak menghasilkan *inkonsistensi* data yang mengakibatkan ketergantungan fungsional yang sudah benar menjadi tidak terpenuhi. Akan tetapi, dalam upaya untuk memelihara ketergantungan fungsional yang ada untuk tetap terpenuhi tersebut, prosesnya harus dapat dilakukan dengan efisien.

Contoh :

Tabel mahasiswa : (nim, nama_mhs, alamat_mhs, tgl_lahir) dengan ketergantungan fungsional–nya yaitu :

nim → nama_mhs, alamat_mhs, tgl_lahir

Tabel nilai : (nama_kul, nim, nama_mhs, indeks_nilai) dengan ketergantungan fungsional–nya yaitu :

nama_kul, nim → indeks_nilai dan

nim → nama_mhs

Jika ada perubahan nama_mhs di tabel mahasiswa maka perubahan tersebut harus juga dilakukan di tabel nilai dan juga berlaku sebaliknya yaitu jika ada perubahan nama_mhs di tabel nilai maka perubahan tersebut harus juga dilakukan di tabel mahasiswa.

Jika hal tersebut tidak dilakukan maka data menjadi tidak konsiste dan ketergantunan fungsional menjadi tidak terpenuhi. Misalnya nilai yang nim–nya sama tetapi nama_mhs berbeda. Jika begitu, maka perubahan harus dilakukan diseluruh basis data, tetapi masalahnya perubahan itu tidak efisien dan seharusnya dihindari.Karena itu sebaiknya agar kriteria *dependency preservation* dapat terpenuhi dengan meniadakan/melepaskan atribut nama_mhs dari tabel nilai (sehingga tabel nilai berisi 3 atribut, yaitu nama-kul, nim dan indeks_nilai).

**8) *Boyce Code Normal Form (BCNF)***

Kriteria berikutnya untuk mendapatkan tabel yang baik adalah dengan menerapkan BCNF. Sebuah tabel dikatakan memenuhi BCNF jika untuk semua ketergantungan fungsional dengan notasi X → Y, maka X harus merupakan *candidate key* pada tabel tersebut. Jika tidak demikian, maka tabel tersebut harus didekomposisi berdasarkan ketergantungan fungsional yang ada, sedemikian hingga X menjadi *candidat key* dari tabel–tabel hasil dekomposisi.

*Contoh tabel yang tidak memenuhi BCNF :*

Ditentukan tabel A = (E, F, G, H, I) dan berlaku ketergantungan fungsional, yaitu :

E, F → G, H, I

F, G → H, I

Disini tabel A tidak memenuhi BCNF karena ada X yang bukan *candiday key*, yaitu F, G sehingga F, G → H, I.

Sedangkan E, F adalah *candidat key* karena E, F → G, H, I

Karena terdapat 2 ketergantungan fungsional maka agar tabel A tidak memenuhi BCNF maka tabel tersebut harus didekomposisikan menjadi :

$A_1$ = (E, F, G) dengan ketergantungan fungsional E, F → G

$A_2$ = (F, G, H, I) dengan ketergantungan fungsional F, G → H, I

**c. Rangkuman**

Perancangan basis data merupakan suatu hal yang sangat penting. Tujuan perancangan adalah agar dapat memiliki basis data yang kompak, efisien dalam penggunaan ruang penyimpanan, cepat dalam pengaksesan dan mudah dalam memanipulasi. Perancangan basis data meliputi perancangan model konseptual dan model phisik. Dalam merancang basis data dapat dilakukan melalui dua pendekatan yaitu: 1) Model Entity–Relationship-diagram dan 2) Menerapkan normalisasi terhadap struktur tabel yang telah diketahui.

Normalisasi diartikan sebagai suatu teknik yang menstrukturkan atau mendekomposisi atau memecah data menggunakan cara–cara tertentu untuk mencegah timbulnya permasalahan pengolahan data dalam basis data. Beberapa konsep dasar yang berkaiatan dengan normalisasi data: 1) ketergantungan fungsional (sudah dibahas dalam kegiatan belajar tujuh). 2) Domain dan tipe data. 3) Konsep key atribut (Field/atribute kunci).

Domain memiliki kesamaan arti dengan fungsi tipe data. Tipe data merujuk pada kemampuan penyimpanan data yang mungkin bagi suatu atribut secara fisik, tanpa melihat layak atau tidaknya data bila dilihat dari kenyataannya pemakaiannya. Domain nilai lebih ditetapkan pada batas–batas nilai yang diperbolehkan bagi suatu atribut, dilihat dari kenyataanya yang ada. *Key* atribut adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data (row) dalam table secara unik. Ragam attribute kunci antara lain :*Candidate Key* (Kunci Kandidat/Kunci Calon), *Primary Key* (Kunci Primer), *Alternate Key* (Kunci Alternatif), *Foreign Key* (Kunci Tamu)

Dalam perspektif normalisasi, sebuah tabel dapat dikategorikan baik (*efisien*) atau normal, jika telah memenuhi 3 (tiga) kriteria yaitu: 1) Jika ada *dekomposisi* (penguraian) tabel, maka dekomposisinya harus dijamin aman (*Lossless–Join Decomposition*). 2) Terpeliharanya ketergantungan fungsional pada saat perubahan data (*DependencyPreservation*). 3) Tidak melanggar *Boyce–Code Normal Form* (BCNF).

**d. Tugas : Mengoperasikan Teknik Normalisasi Data**

Dalam kegiatan ini peserta didik akan melakukan eksperimen atau praktikum secara berkelompok, satu kelompok terdiri dari dua sampai tiga orang. Dalam Eksperimen ini akan dilakukan pengujian terhadap *entity relationship diagram*

yang telah dibuat dalam tugas kegiatan belajar 5. Pengujian yang dilakukan merujuk pada dua kriteria yaitu (*Lossless–Join Decomposition* dan *Dependency Preservation.* Bacalah seluruh langkah eksperimen dibawah ini kemudian lakukan dengan cermat dan teliti dengan perangkat yang telah disediakan.

1. Lihat dan amati kembali hasil tugas belajar kegiatan 5 tentang pemetaan ER diagram ke relasi tabel.
2. Pastikan dalam relasi tabel diatas terdapat relasi one to one, relasi one to many, relasi many to many dan relasi dan relasi ternary. Jika belum ada salah satu jenis relasi tersebut buat relasi tabelnya.
3. Untuk setiap tabel dalam langkah dua di atas lengkapi data dengan menambahkan *record-record* dengan jumlah record 3 -5 record. Tampilkan hasilnya dalam tabel.
4. Dari hasil tabel pada langkah tiga di atas lakukan pengujian terhadap setiap tabel dalam ERD apakah memenuhi persyaratan *Lossless–Join Decomposition* .Tampilkan hasilnya dalam tabel.
5. Ulangi langkah kegiatan 4 (empat) dengan kriteria persyaratan *Dependency Preservation.* Tampilkan hasilnya dalam tabel.
6. Kumpulkan data-data setiap langkah dan analisis data tersebut menggunakan analisis diskriptif.
7. Diskusi dan komunikasikan hasilnya dalam kelompok dan buatlah kesimpulan.
8. Buatlah Laporan dan komunikasikan hasil laporan dan pembahasan dengan guru pembimbing.

### e. Test Formatif.

Dalam test ini setiap peserta didik membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

1. Jelaskan secara singkat definisi dan manfaat teknik normalisasi data?
2. Sebutkan dan jelaskan tiga konsep dasar yang dibutuhkan dalam teknik normalisasi data ?
3. Sebutkan dan jelaskan tiga persyaratan dalam teknik normalisasi data ?

**f. Lembar Jawaban Test Formatif (LJ).**

**LJ- 01 : Pengertian dan manfaat (kegunaan) teknik normalisasi data.**

**LJ- 02 : Tiga Konsep dasar yang diperlukan dalam teknik normalisasi data?**

g. **Lembar Kerja Peserta Didik.**

## Kegiatan belajar 9: Tahapan Proses Normalisasi.

**a. Tujuan Pembelajaran.**

Setelah mengikuti kegiatan belajar 9 ini diharapkan peserta didik dapat:

- ✓ Memahami tahapan proses normalisasi data.
- ✓ Memahami bentuk-bentuk normalisasi data.
- ✓ Menguji tabel relasional menggunakan bentuk normal tahap 1 (1st NF).
- ✓ Menguji tabel relasional menggunakan bentuk normal tahap 2 (2nd NF).
- ✓ Menguji tabel relasional menggunakan bentuk normal tahap 3 (3rd NF).

**b. Uraian materi.**

**1) Bentuk-Bentuk Normalisasi**

Normalisasi data adalah proses yang berkaitan dengan model data relasional untuk mengorganisasi himpunan data dengan ketergantungan dan keterkaitan yang tinggi atau erat. Hasil dari proses normalisasi adalah tabel–tabel data dalam bentuk normal (*normal form*), yaitu tabel–tabel data yang terhindar dari dua hal yaitu:

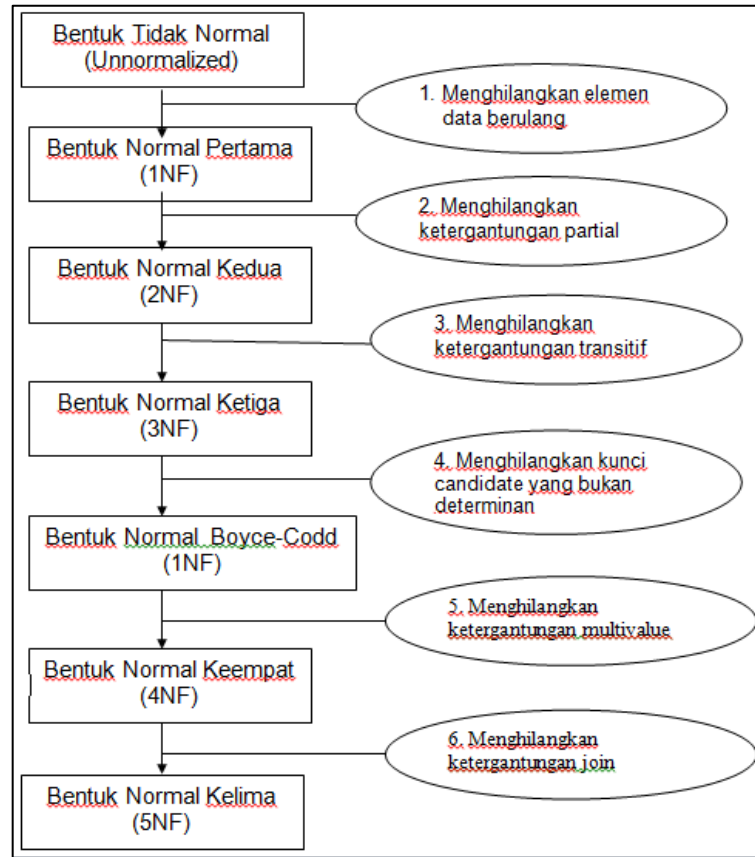
- Pengulangan informasi.
- Potensi *inkonsistensi* data pada operasi pengubahan.

Terdapat enam bentuk normal (normal form) dalam teknik normalisasi data, keenam bentuk tersebut adalah :

1. Bentuk Normal Tahap pertama (1st Normal Form)
2. Bentuk Normal Tahap Kedua (2nd Normal Form)
3. Bentuk Normal Tahap Ketiga (3rd Normal Form)
4. Bentuk Normal Boyce - Code (BCNF)
5. Bentuk Normal Tahap Keempat (4rd Normal Form)
6. Bentuk Normal Tahap Kelima (4rd Normal Form)

**2) Proses-Proses Normalisasi data**

Dalam proses normalisasi, data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat. Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk

yang optimal. Langkah-langkah yang dilakukan dalam melakukan normalisasi data diperlihatkan dalam gambar dibawah ini:

**Gambar 42. Langkah-langkah proses normalisasi data**

### 3) Bentuk tidak normal (Unnormalized Form)

Bentuk ini memiliki ciri-ciri, yaitu :

- Merupakan kumpulan data yang akan direkam
- Tidak ada keharusan mengikuti suatu format tertentu
- Dapat saja data tidak lengkap atau terduplikasi
- Data dikumpulkan apa adanya sesuai dengan kedatangannya.

### 4) Bentuk Normal Tahap pertama (1st Normal Form)

Bentuk normal ke satu 1 NF ini mempunyai beberapa ciri antara lain yaitu:

- Setiap data dibentuk dalam flat file (file data/ rata)
- Data dibentuk dalam satu record demi satu record dan nilai dari field field berupa "atomic value", tidak dapat dibagi-bagi lagi.

- Tidak ada set atribute yang berulang ulang atau atribute bernilai ganda (multivalue).
- Tidak ada set atribut *composite* atau kombinasinya dalam domain data yang sama.
- Tiap field hanya satu pengertian, bukan merupakan kumpulan kata yang mempunyai arti mendua, hanya satu arti saja dan juga bukanlah pecahan kata sehingga artinya lain.

Contoh dari tabel yang belum memenuhi 1NF :

| nis | namasiswa | hobi |
|---|---|---|
| 111 | Tiara Hadira | Memasak, Nonton, shopping |
| 112 | Hadi Saputra | Programming, game online, Memancing |
| 113 | Rima Aninda | Membaca buku, bulu tangkis |
| 114 | Nindia Sari | Membuat kue, membaca novel, golf |
| 115 | Saputra Sinara | Memancing, travelling, sepakbola |
| 116 | Diana Riani Tiara | Membaca buku, memasak, bowling |
| 117 | Tora Hadira Putra | Sepakbola, tennis, renang |

Gambar 43. Contoh data pada tabel yang belum memenuhi 1NF

Atau bentuk entitas seperti berikut :

| nis | namasiswa | hobi1 | hobi2 | hobi3 |
|---|---|---|---|---|
| 111 | Tiara Hadira | Memasak | Nonton | shopping |
| 112 | Hadi Saputra | Programming komputer | game online | Memancing |
| 113 | Rima Aninda | Membaca buku | bulu tangkis | |
| 114 | Nindia Sari | Membuat kue | membaca novel | golf |
| 115 | Saputra Sinara | Memancing | travelling | sepakbola |
| 116 | Diana Riani Tiara | Membaca buku | memasak | bowling |
| 117 | Tora Hadira Putra | Sepakbola | tennis | renang |

Gambar 44. Contoh lain dari data pada tabel lain yang belum memenuhi 1NF

Untuk dapat memenuhi aturan 1NF, maka dilakukan penataan ulang data (dekomposisi) menjadi 2 entitas, yakni entitas siswa dan entitas hobi seperti gambar berikut :

Tabel siswa

| nis | namasiswa |
|---|---|
| 111 | Tiara Hadira |
| 112 | Hadi Saputra |
| 113 | Rima Aninda |
| 114 | Nindia Sari |
| 115 | Saputra Sinara |
| 116 | Diana Riani Tiara |
| 117 | Tora Hadira Putra |

Tabel hobi

| nis | hobi |
|---|---|
| 111 | Shopping |
| 115 | Sepakbola |
| 117 | Sepakbola |
| 112 | Programming |
| 111 | Nonton |
| 114 | Membuat kue |
| 114 | Membaca novel |
| 113 | Membaca buku |
| 116 | Membaca buku |
| 116 | memasak |
| 111 | Memasak |
| 115 | Memancing |
| 112 | Memancing |
| 114 | Golf |
| 112 | Game online |
| 113 | Bulu tangkis |
| 115 | Travelling |
| 117 | tennis |
| 117 | renang |

Gambar 45. Hasil dekomposisi tabel untuk memenuhi bentuk 1NF

### 5) Bentuk Normal Tahap Kedua (2nd Normal Form)

Bentuk normal kedua mempunyai syarat yaitu:

- Bentuk data telah memenuhi kriteria bentuk normal kesatu.
- Atribute bukan kunci haruslah bergantung secara fungsi pada kunci utama atau primary key.
- Sudah ditentukan kunci kunci field, dimana kunci field haruslah unik dan dapat mewakili atribute lain yang menjadi anggotanya.

Sebagai contoh ditentukan sebuah tabel siswa sebagai berikut :

| <u>NIS</u> | Nama_siswa | Alamat | Kode_ Mapel | Nama_Mapel | Nama_Guru | Nilai |
|---|---|---|---|---|---|---|

Tabel di atas telah memenuhi 1NF, namun belum memenuhi 2NF, {NIS, Kode_Mapel} yang dianggap sebagai primary key sedangkan:

{NIS, Kode_Mapel} $\nrightarrow$ Nama_siswa
{NIS, Kode_Mapel} $\nrightarrow$ Alamat
{NIS, Kode_Mapel} $\nrightarrow$ Nama_Mapel
{NIS, Kode_Mapel} $\nrightarrow$ Nama_Guru
{NIS, Kode_Mapel} $\rightarrow$ nilai (tergantung secara fungsi atau *functional dependency*)

Tabel di atas perlu didekomposisi menjadi beberapa tabel untuk memenuhi syarat 2NF. Dekomposisi sesuai dengan *functionaldependencynya* (FD) adalah sebagai berikut :

FD 1 : {NIS, Kode_Mapel} → Nilai

FD 2 : NIS → {Nama_siswa, Alamat}

FD 3 : Kode_mapel →{Nama_mapel, Nama_guru}

Dari ketiga FD di atas, maka dilakukan dekomposisi tabel menjadi sebagai berikut :

Tabel Nilai : (NIS, Kode_mapel, Nilai)

Tabel Siswa :(NIS, Nama_siswa, Alamat)

Tabel Mapel :(Kode_mapel, Nama_mapel, Nama_Guru)

**6) Bentuk Normal Tahap Ketiga (3rd Normal Form)**

Untuk menjadi bentuk normal ketiga (3 NF) suatu tabel harus mempunyai ciri-ciri sebagai berikut:

1. Memenuhi bentuk 2 NF (normal kedua)
2. Atribut bukan kunci tidak memiliki dependensi transitif terhadap kunci utama atau primary key.
3. Setiap atribute bukan kunci haruslah bergantung hanya pada primary key dan pada primary key secara menyeluruh

Berikut ini adalah contoh relasi yang telah memenuhi bentuk 2 NF, tetapi belum memenuhi bentuk 3 NF :

| NIS | Nama_siswa | Alamat_jln | Alamat_kota | Alamat_prov | Kodepos |
|---|---|---|---|---|---|

Pada relasi di atas, masih terdapat atribut non primary key (yakni Alamat_kota dan Alamat_Prov) yang memiliki ketergantungan terhadap atribut non primary key yang lain, yaitu Kode_pos.

Kodepos → {Alamat_kota, Alamat_prov}

Untuk memenuhi syarat 3NF, maka relasi tersebut harus didekomposisi sebagai berikut :

Siswa : (NIS, Nama_siswa, Alamat_jn, Kodepos)

Kodepos : (Kodepos, Alamat_kota, Alamat_prov)

### c. Rangkuman

Normalisasi data adalah proses yang berkaitan dengan model data relasional untuk mengorganisasi himpunan data dengan ketergantungan tinggi. Hasil dari proses normalisasi adalah tabel data dalam bentuk normal. Terdapat enam bentuk normal tabel yaitu: 1) Bentuk Normal Tahap pertama ($1^{st}$ NF). 2) Bentuk Normal Tahap Kedua ($2^{nd}$ NF). 3) Bentuk Normal Tahap Ketiga ($3^{rd}$ NF). 4) Bentuk Normal Boyce - Code (BCNF). 5) Bentuk Normal Tahap Keempat ($4^{rd}$ NF). 6) Bentuk Normal Tahap Kelima ($4^{rd}$ NF)

Bentuk normal ke satu 1 NF ini mempunyai beberapa ciri: 1) Setiap data dibentuk dalam flat file (file data). 2) Data dibentuk dalam satu record demi satu record dan nilai dari field field berupa "atomic value", tidak dapat dibagi-bagi lagi. 3) Tidak ada set atribute yang berulang ulang atau atribute bernilai ganda (multivalue). 4) Tidak ada set atribut *composite* atau kombinasinya dalam domain data yang sama.

Bentuk normal kedua mempunyai syarat yaitu: 1) Bentuk data telah memenuhi kriteria bentuk normal kesatu. 2) Atribute bukan kunci haruslah bergantung secara fungsi pada kunci utama atau primary key. 3) Sudah ditentukan kunci kunci field, dimana kunci field haruslah unik dan dapat mewakili atribute lain yang menjadi anggotanya.

Untuk menjadi bentuk normal ketiga (3 NF) suatu tabel harus mempunyai ciri-ciri sebagai berikut: 1) Memenuhi bentuk 2 NF (normal kedua). 2) Atribut bukan kunci tidak memiliki dependensi transitif terhadap kunci utama atau primary key. 3) Setiap atribute bukan kunci haruslah bergantung hanya pada primary key dan pada primary key secara menyeluruh.

Langkah-langkah yang dilakukan dalam melakukan normalisasi data adalah : 1) menghilangkan elemen data berulang. 2) menghilangkan ketergantungan parsial. 3) menghilangkan ketergantungan transitif. 4) menghilangkan kunci kandidiat yang bukan determinan.5) menghilangkan ketergantungan multi value. 6) menghilangkan ketergantungan join.

.

**d. Tugas : Mengoperasikan Teknik Normalisasi data**

Dalam kegiatan ini peserta didik akan melakukan eksperimen atau praktikum secara berkelompok, satu kelompok terdiri dari dua sampai tiga orang. Dalam Eksperimen ini akan dilakukan pengujian terhadap *entity relationship diagram*yang telah dibuat dalam tugas kegiatan belajar 5. Pengujian yang dilakukan merujuk pada teknik normalisasi data bentuk normal 1(1$^{st}$ NF)., bentuk normal 2 (2$^{nd}$ NF)dan bentuk normal 3 (3$^{rd}$ NF).Bacalah seluruh langkah eksperimen dibawah ini kemudian lakukan dengan cermat dan teliti dengan perangkat yang telah disediakan.

1. Lihat dan amati kembali hasil tugas belajar kegiatan 5 tentang pemetaan ER diagram ke relasi tabel.
2. Pastikan dalam relasi tabel diatas terdapat relasi one to one, relasi one to many, relasi many to many dan relasi dan relasi ternary. Jika belum ada salah satu jenis relasi tersebut buat relasi tabelnya.
3. Untuk setiap tabel dalam langkah dua di atas lengkapi data dengan menambahkan *record-record* dengan jumlah record 3 -5 record. Tampilkan hasilnya dalam tabel.
4. Dari hasil tabel pada langkah tiga di atas lakukan pengujian terhadap setiap tabel dalam ERD, apakah memenuhi persyaratan bentuk normal 1 (1$^{st}$ NF). Tampilkan hasilnya dalam tabel atau gambar.
5. Ulangi langkah 4 (empat) untuk bentuk normal 2(2$^{nd}$ NF) Tampilkan hasilnya dalam tabel atau gambar.
6. Ulangi langkah 4 (empat) untuk bentuk normal 3 (3$^{rd}$ NF). Tampilkan hasilnya dalam tabel atau diagram gambar.
7. Kumpulkan data-data setiap langkah dan analisis data tersebut menggunakan analisis diskriptif.
8. Diskusi dan komunikasikan hasilnya dalam kelompok dan buatlah kesimpulan.
9. Buatlah Laporan dan komunikasikan hasil laporan dan pembahasan dengan guru pembimbing.

**e. Test Formatif.**

Dalam test ini setiap peserta didik membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

1. Jelaskan secara singkat definisi teknik normalisasi datadan hasil yang didapat dari proses normalisasi?
2. Jelaskan proses-proses normalisasi data ?
3. Jelaskan secara singkat bentuk normalisasi data 1 NF, 2 NF dan 3NF

**f. Lembar Jawaban Test Formatif (LJ).**

**LJ- 01 : Pengertian Normalisasi data dan hasil yang didapat..**

................................................................................................................
................................................................................................................
................................................................................................................
................................................................................................................
................................................................................................................
................................................................................................................
................................................................................................................
................................................................................................................
................................................................................................................
................................................................................................................
................................................................................................................
................................................................................................................
................................................................................................................
................................................................................................................
................................................................................................................
................................................................................................................

**LJ- 02 : Proses-Proses Normalisasi Data ?**

................................................................................................................
................................................................................................................
................................................................................................................

## Kegiatan belajar 10 : Tahab proses Normalisasi 2

**a. Tujuan Pembelajaran.**

Setelah mengikuti kegiatan belajar10 ini diharapkan peserta didik dapat:

- ✓ Memahami tahapan proses normalisasi data.
- ✓ Memahami bentuk-bentuk normalisasi data.
- ✓ Menguji tabel relasional menggunakan *Boyce Code Normal Form (BCNF)*
- ✓ Menguji tabel relasional menggunakan bentuk normal tahap 4 ($4^{nd}$ NF).
- ✓ Menguji tabel relasional menggunakan bentuk normal tahap 5 ($5^{rd}$ NF).

**b. Uraian materi.**

**1) Bentuk-Bentuk Normalisasi**

Normalisasi data adalah proses yang berkaitan dengan model data relasional untuk mengorganisasi himpunan data dengan ketergantungan dan keterkaitan yang tinggi atau erat. Hasil dari proses normalisasi adalah tabel–tabel data dalam bentuk normal (*normal form*), yaitu tabel–tabel data yang terhindar dari dua hal yaitu:

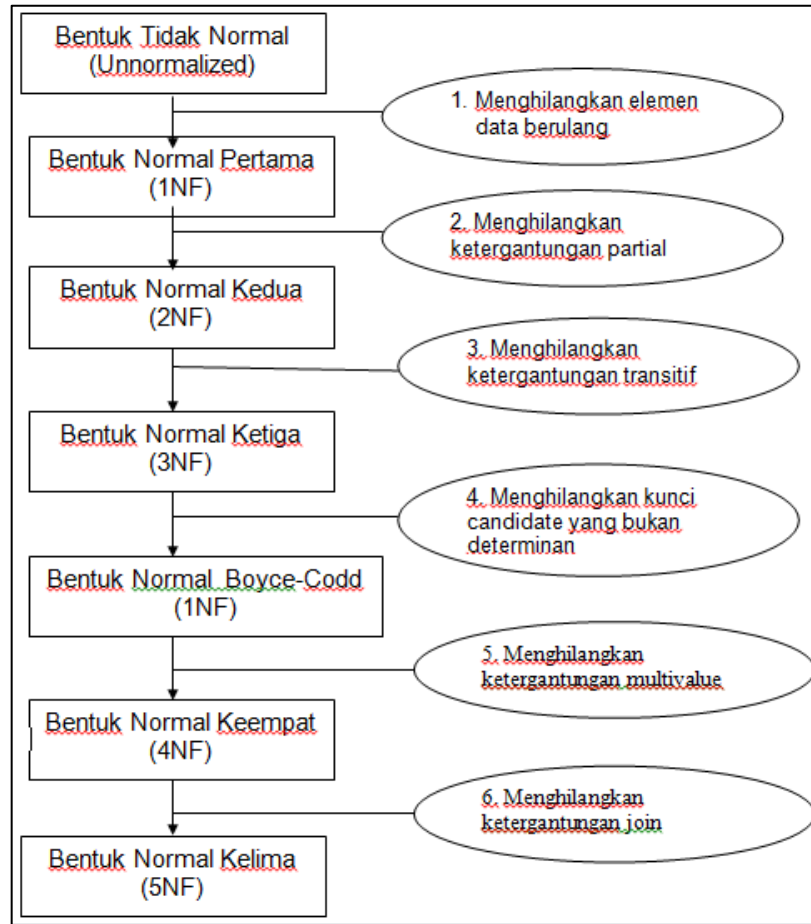
- Pengulangan informasi.
- Potensi *inkonsistensi* data pada operasi pengubahan.

Terdapat enam bentuk normal (normal form) dalam teknik normalisasi data, keenam bentuk tersebut adalah :

1. Bentuk Normal Tahap pertama ($1^{st}$ Normal Form)
2. Bentuk Normal Tahap Kedua ($2^{nd}$ Normal Form)
3. Bentuk Normal Tahap Ketiga ($3^{rd}$ Normal Form)
4. Bentuk Normal Boyce - Code (BCNF)
5. Bentuk Normal Tahap Keempat ($4^{rd}$ Normal Form)
6. Bentuk Normal Tahap Kelima ($5^{rd}$ Normal Form)

**2) Proses-Proses Normalisasi data**

Dalam proses normalisasi, data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat. Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk

yang optimal. Langkah-langkah yang dilakukan dalam melakukan normalisasi data diperlihatkan dalam gambar dibawah ini:

**Gambar 46. Langkah-langkah normalisasi data**

### 3) *Boyce Code Normal Form (BCNF)*

BCNF merupakan bentuk normal sebagai perbaikan terhadap 3NF. Suatu relasi yang memenuhi BCNF selalu memenuhi 3NF, tetapi tidak untuk sebaliknya. Suatu relasi yang memenuhi 3NF belum tentu memenuhi BCNF. Karena dalam bentuk 3 NF masih memungkinkan terjadi anomali.

Sebuah tabel dikatakan memenuhi BCNF jika untuk semua ketergantungan fungsional dengan notasi X → Y, maka X harus merupakan *candidate key* pada tabel tersebut. Jika tidak demikian, maka tabel tersebut harus didekomposisi berdasarkan ketergantungan fungsional yang ada, sedemikian hingga X menjadi *candidat key* dari tabel–tabel hasil dekomposisi.

*Contoh tabel yang tidak memenuhi BCNF :*

Ditentukan tabel A = (E, F, G, H, I) dan berlaku ketergantungan fungsional, yaitu :

E, F → G, H, I

F, G → H, I

Disini tabel A tidak memenuhi BCNF karena ada X yang bukan *candiday key*, yaitu F, G sehingga F, G → H, I.

Sedangkan E, F adalah *candidat key* karena E, F → G, H, I

Karena terdapat 2 ketergantungan fungsional maka agar tabel A tidak memenuhi BCNF maka tabel tersebut harus didekomposisikan menjadi :

$A_1$ = (E, F, G) dengan ketergantungan fungsional E, F → G

$A_2$ = (F, G, H, I) dengan ketergantungan fungsional F, G → H, I

Contoh lain untuk bentuk ini adalah tabel SEMINAR berikur, kunci primernya adalah no_peserta dan kode_seminar, dengan asumsi bahwa :

Peserta dapat mengambil 1 atau 2 seminar.

Setiap seminar membutuhkan 2 instruktur.

Setiap peserta dibimbing oleh salah satu dari 2 instruktur seminar.

Setiap instruktur boleh hanya membimbing 1 seminar saja.

Pada contoh relasi berikut, no_peserta dan kode_seminar menunjukkan seorang instruktur.

| No_Peserta | Kode_seminar | Nama_instruktur |
|---|---|---|
| 2201001 | 2281 | Santi |
| 2201002 | 2281 | Karyadi |
| 2201003 | 2291 | Jeni |
| 2201002 | 2291 | Rendi |
| 2201004 | 2291 | Rendi |

Bentuk relasi SEMINAR adalah memenuhi bentuk normal ketiga (3NF), tetapi tidak BCNF karena Kode_seminar masih bergantung fungsi pada instruktur, jika setiap instruktur dapat mengajar hanya pada satu seminar. Kode_seminar bergantung fungsi pada satu atribut bukan superkey seperti yang disyaratkan oleh BCNF. Maka relasi SEMINAR harus didekomposisi menjadi dua relasi, yaitu relasi pengajar dan seminar_instruktur, seperti berikut ini :

Pengajar : (Nama_instruktur, Kode_seminar) dan

Seminar_instruktur : (No_peserta, Nama_instruktur)

**4) Bentuk Normal Tahap ke empat (4th Normal Form)**

Suatu tabel relasional dikatakan dalam bentuk normal keempat (4NF) jika memenuhi beberapa ketentuan sebagai berikut : Bila dan hanya bila telah berada dalam bentuk BCNF dan tidak adamultivalued dependency nontrivial.

Multivalued dependency (MVD) dipakai dalam bentuk normal keempat (4NF). Dependensi ini dipakai untuk menyatakan hubungan satu ke bantak (one tomany).

Setiap atribut di dalamnya tidak mengalami ketergantungan pada banyak nilai atau dengan kalimat lain, bahwa semua atribut yang mengalami ketergantungan pada banyak nilai adalah bergantung secara fungsional (*functionally dependency*)

Berikut ini adalah salah satu contoh tabel relasi0nal yang belum memenuhi 4NF :

| Matakuliah | Dosen | Isi |
|---|---|---|
| Pengenalan Komputer | Budi | Dasar Komputer |
| Pengenalan Komputer | Budi | Pengenalan pengolahan kata |
| Pengenalan Komputer | Budi | Pengenalan lembaran kerja |
| Pengenalan Komputer | Sanjaya | Dasar Komputer |
| Pengenalan Komputer | Sanjaya | Pengenalan pengolahan kata |
| Pengenalan Komputer | Sanjaya | Pengenalan lembaran kerja |
| Matematika | Sugeng Paijo | Differensial |
| Matematika | Sugeng Paijo | Integral |

Relasi tersebut menggambarkan mengenai dosen yang mengajar matakuliah tertentu dengan isi matakuliah yang bersangkutan. Contoh tabel dibawah ini menjelaskan dua dosen yang mengajar pengenalan komputer, yaitu Budi dan Sanjaya.

| Matakuliah | Dosen | Isi |
|---|---|---|
| Pengenalan Komputer | Budi<br>Sanjaya | Dasar Komputer<br>Pengenalan pengolahan kata<br>Pengenalan lembaran kerja |
| Matematika | Sugeng Paijo | Differensial<br>Integral |

Adapun isi matakuliah Pengenalan Komputer adalah Dasar Komputer, Pengenalan Pengolahan Kata dan Pengenalan Lembaran Kerja. Relasi berikut

ini memperlihatkan relasi yang telah dinormalisasikan berdasarkan relasi sebelumnya.Langkah selanjutnya guna memenuhi syarat bentuk normal tahap 4), maka relasi tersebut diatas dapat didekomposisi menjadi dua relasi sebagai berikut :

Matakuliah_dosen : (Matakuliah, Dosen)

Matakuliah_isi : (Matakuliah, Isi)

**5) *Bentuk Normal Tahap Kelima (5$^{th}$ Normal Form)***

Bentuk normal 5NF terpenuhi jika tidak dapat memiliki sebuah *lossless decomposition* menjadi tabel-tabel yg lebih kecil. Jika 4 bentuk normal sebelumnya dibentuk berdasarkan *functional dependency*, 5NF dibentuk berdasarkan konsep *join dependence*. Yakni apabila sebuah tabel telah di-dekomposisi menjadi tabel-tabel lebih kecil, harus bisa digabungkan lagi (join) untuk membentuk tabel semula, sehingga bentuk normal kelima disebut juga sebagai *Projection Join Normal Form* (PJNF).Suatu tabel memenuhi bentuk normal 5$^{rd}$NF jika dan hanya jika Kerelasian antar data dalam relasi tersebut tidak dapat direkonstruksi dari struktur relasi yang memuat atribut yang lebih sedikit.

Sebagai contoh: terdapat hubungan dealer yaitu suatu perusahaan distributor kendaraan. Dalam hal ini distributor memiliki sejumlah produk kendaraan. Tabel relasional dibawah ini menjelaskan relasi tabel dealer, kendaraan dan distributor.

| Dealer | Distributor | Kendaraan |
|---|---|---|
| Sumber Jaya | Nissan | Truk Nissan |
| Sumber Jaya | Toyota | Toyota Kijang |
| Sumber Jaya | Toyota | Truk Dyna |
| Asterindo | Nissan | Sedan Nissan |

Relasi tersebut telah memenuhi dependensi gabungan, Sehingga relasi tersebut dapat didekomposisi menjadi tiga buah relasi yaitu :

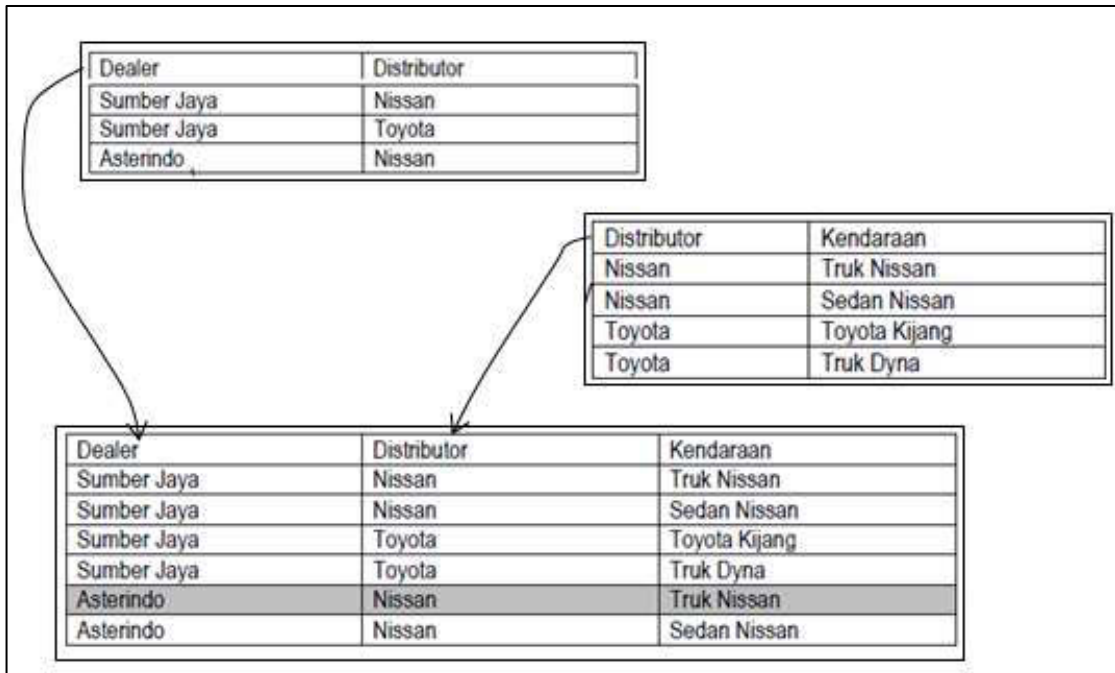
- Deal_Dist (Dealer_Distributor).
- Dist_Kend (Distributor_Kendaraan).
- Deal_Kend (Dealer_Kendaraan).

Gabungan ketiga relasi tersebut akan membentuk relasi Dealer-Distributor-Kendaraan (DDK) dan gabungan ketiganya. Kemungkinan proyeksitabel

relasional tersebutakan menghasilkan suatu*relasi antara* yang salah, namum ketiganya akan menghasilkan relasi sesuai aslinya. Gambar dibawah ini menjelaskan tabel relasional Dealer-Distributor-Kendaraan (DDK)

| Dealer | Distributor |
|---|---|
| Sumber Jaya | Nissan |
| Sumber Jaya | Toyota |
| Asterindo | Nissan |

| Distributor | Kendaraan |
|---|---|
| Nissan | Truk Nissan |
| Nissan | Sedan Nissan |
| Toyota | Toyota Kijang |
| Toyota | Truk Dyna |

| Dealer | Distributor | Kendaraan |
|---|---|---|
| Sumber Jaya | Nissan | Truk Nissan |
| Sumber Jaya | Nissan | Sedan Nissan |
| Sumber Jaya | Toyota | Toyota Kijang |
| Sumber Jaya | Toyota | Truk Dyna |
| Asterindo | Nissan | Truk Nissan |
| Asterindo | Nissan | Sedan Nissan |

Gambar 47. Tabel relasional Dealer-Distributor-Kendaraan (DDK)

**6) Efek Normalisasi**

Pada kenyataannya, penerapan normalisasi juga mengakibatkan efek samping yang tidak diharapkan,yaitu :

1. Proses dekomposisi relasi akan mengakibatkan munculnya duplikasi rinci data pada atribut kunci penghubung (foreign key).
2. Dekomposisi relasi membuka kemungkinan tidak terpenuhi integritas refernsial (referential integrity) dalam basis data.
3. Dekomposisi relasi akan menghasilkan semakin banyak jumpak relasi baru, sehingga mengakibatkan inefisiensi proses menampilkan kembali data-data dari dalam basis data.
4. Adanya batasan penerapan pada beberapa DBMS untuk ukuran computer pribadi/PC, berkaitan dengan batas maksimal relasi yang dapat dibuka secara bersamaan.

**c. Rangkuman**

Normalisasi data adalah proses yang berkaitan dengan model data relasional untuk mengorganisasi himpunan data dengan ketergantungan tinggi. Hasil dari proses normalisasi adalah tabel data dalam bentuk normal. Terdapat enam bentuk normal tabel yaitu: 1) Bentuk Normal Tahap pertama (1$^{st}$ NF). 2) Bentuk Normal Tahap Kedua (2$^{nd}$ NF). 3) Bentuk Normal Tahap Ketiga (3$^{rd}$ NF). 4) Bentuk Normal Boyce - Code (BCNF). 5) Bentuk Normal Tahap Keempat (4$^{rd}$ NF). 6) Bentuk Normal Tahap Kelima (4$^{rd}$ NF)

BCNF merupakan bentuk normal sebagai perbaikan terhadap 3NF. Suatu relasi yang memenuhi BCNF selalu memenuhi 3NF, tetapi tidak untuk sebaliknya. Suatu relasi yang memenuhi 3NF belum tentu memenuhi BCNF. Karena dalam bentuk 3 NF masih memungkinkan terjadi anomali

Suatu tabel relasional dikatakan dalam bentuk normal keempat (4NF) jika memenuhi beberapa ketentuan sebagai berikut : 1) Bila dan hanya bila telah berada dalam bentuk BCNF dan tidak ada multivalued dependency nontrivial. 2) Multivalued dependency (MVD) dipakai dalam bentuk normal keempat (4NF). 3) Dependensi ini dipakai untuk menyatakan hubungan one to manyBentuk normal 5NF terpenuhi jika tidak dapat memiliki sebuah *lossless decomposition* menjadi tabel-tabel yg lebih kecil. Jika 4 bentuk normal sebelumnya dibentuk berdasarkan *functional dependency*, 5NF dibentuk berdasarkan konsep *join dependence.*

Penerapan normalisasi mengakibatkan efek samping yang tidak diharapkan,yaitu :1) Proses dekomposisi relasi akan mengakibatkan munculnya duplikasi rinci data pada atribut kunci penghubung (foreign key). 2) Dekomposisi relasi membuka kemungkinan tidak terpenuhi integritas refernsial (referential integrity) dalam basis data. 3) Dekomposisi relasi akan menghasilkan semakin banyak jumpak relasi baru, sehingga mengakibatkan inefisiensi proses menampilkan kembali data-data dari dalam basis data. 4) Adanya batasan penerapan pada beberapa DBMS untuk ukuran computer pribadi/PC, berkaitan dengan batas maksimal relasi yang dapat dibuka secara bersamaan

.

**d. Tugas : Mengoperasikan Teknik Normalisasi data**

Dalam kegiatan ini peserta didik akan melakukan eksperimen atau praktikum secara berkelompok, satu kelompok terdiri dari dua sampai tiga orang. Dalam Eksperimen ini akan dilakukan pengujian terhadap *entity relationship diagram*

yang telah dibuat dalam tugas kegiatan belajar 5. Pengujian yang dilakukan merujuk pada teknik normalisasi data bentuk BCNF, bentuk normal 4 ($4^{th}$NF)dan bentuk normal 5 ($4^{rth}$NF).Bacalah seluruh langkah eksperimen dibawah ini kemudian lakukan dengan cermat dan teliti dengan perangkat yang telah disediakan.

1. Lihat dan amati kembali hasil tugas belajar kegiatan 5 tentang pemetaan ER diagram ke relasi tabel.
2. Pastikan dalam relasi tabel diatas terdapat relasi one to one, relasi one to many, relasi many to many dan relasi dan relasi ternary. Jika belum ada salah satu jenis relasi tersebut buat relasi tabelnya.
3. Untuk setiap tabel dalam langkah dua di atas lengkapi data dengan menambahkan *record-record* dengan jumlah record 3 -5 record. Tampilkan hasilnya dalam tabel.
4. Dari hasil tabel pada langkah tiga di atas lakukan pengujian terhadap setiap tabel dalam ERD, apakah memenuhi persyaratan bentuk BCNF. Tampilkan hasilnya dalam tabel atau gambar.
5. Ulangi langkah 4 (empat) untuk bentuk normal 4 ($4^{th}$NF) Tampilkan hasilnya dalam tabel atau gambar.
6. Ulangi langkah 4 (empat) untuk bentuk normal 5($5^{th}$NF). Tampilkan hasilnya dalam tabel atau diagram gambar.
7. Kumpulkan data-data setiap langkah dan analisis data tersebut menggunakan analisis diskriptif.
8. Diskusi dan komunikasikan hasilnya dalam kelompok dan buatlah kesimpulan.
9. Buatlah Laporan dan komunikasikan hasil laporan dan pembahasan dengan guru pembimbing.

**e. Test Formatif.**

Dalam test ini setiap peserta didik membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

1. Jelaskan secara singkat bentuk normalisasi data bentuk BCNF ?
2. Jelaskan secara singkat bentuk normalisasi data bentuk 4 NF ?
3. Jelaskan secara singkat bentuk normalisasi data bentuk 5 NF ?

4. Jelaskan secara singkat Efek Normalisasi ?

**f. Lembar Jawaban Test Formatif (LJ).**

**LJ- 01 : Pengertian Normalisasi data bentuk BCNF.**

.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................

**LJ- 02 : Pengertian Normalisasi data bentuk Normal 4 (4 NF).**

.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................

**Kegiatan belajar 11 : alat bantu pemodelan konseptual data**

**a. Tujuan Pembelajaran.**

Setelah mengikuti kegiatan belajar10 ini diharapkan peserta didik dapat:

- ✓ Memahami konsep pemodelan konseptual data
- ✓ Mengoperasikan tools pemodelan konseptual data menggunakan power designer

**b. Uraian materi.**

**1) Pengenalan alat bantu (tools) Power designer**

PowerDesigner adalah software tools dengan pendekatan model driven yang digunakan untuk untuk menyelaraskan bisnis dan Teknologi informasi. Tools ini merupakan *enterprise modelling* yang membantu mengimplementasikan Enterprise Architecture dan membawa lingkungan manajemen meta-data yang kuat untuk siklus hidup pengembangan aplikasi.

PowerDesigner ini menggabungkan beberapa teknik pemodelan standar seperti UML, Business Process Modeling dan pemodelan data. Dalam implementasinya perangkat lunak ini dapat digunakan secara bersama-sama dengan lingkungan pengembangan terkemuka seperti. NET, Workspace, PowerBuilder, Java, Eclipse. Perangkat lunak ini dapat digunakan untuk membuat model analisis bisnis dan proses desain formal untuk siklus pengembangan perangkat lunak tradisional. Dan bekerja dengan semua RDBMSs modern. PowerDesigner menyediakan satu set unik dari pemodelan perusahaan beberapa tools atau alat yang menyatukan teknik standar dan notasi dari Business Process Modeling, Data Modeling dan pemodelan aplikasi UML dengan fitur canggih lainnya. Tools ini digunakan untuk membantu pengguna dalam menganalisis, merancang, membangun, memelihara aplikasi dengan menggunakan rekayasa perangkat lunak praktis. PowerDesigner adalah tools pemodelan perangkat lunak sistem perusahaan berbasis grafis yang mudah digunakan dan menyediakan beberapa fasilitas antara lain:

- ✓ Pemodelan terintegrasi melalui metodologi dan notasi standar seperti : Data (entity atau relasi), sistem Bisnis (BPMN, BPEL, ebXML), pemodelan aplikasi aplikasi (UML)
- ✓ Pembuatan kode porgam secara otomatis melalui template yang dapat disesuaikan dengan SQL (mendukung lebih dari 50 didukung DBMS) , Java pogramming, framework .NET
- ✓ Kemampuan untuk melakukan *reverse engineering* yang kuat dalam mendokumentasikan dan memperbarui sistem yang ada
- ✓ Merupakan Sebuah solusi repositori perusahaan dengan keamanan yang kuat dan kemampuan untuk membantu pengembangan versi multi-user

**2) Ragam model dalam Power Designer**

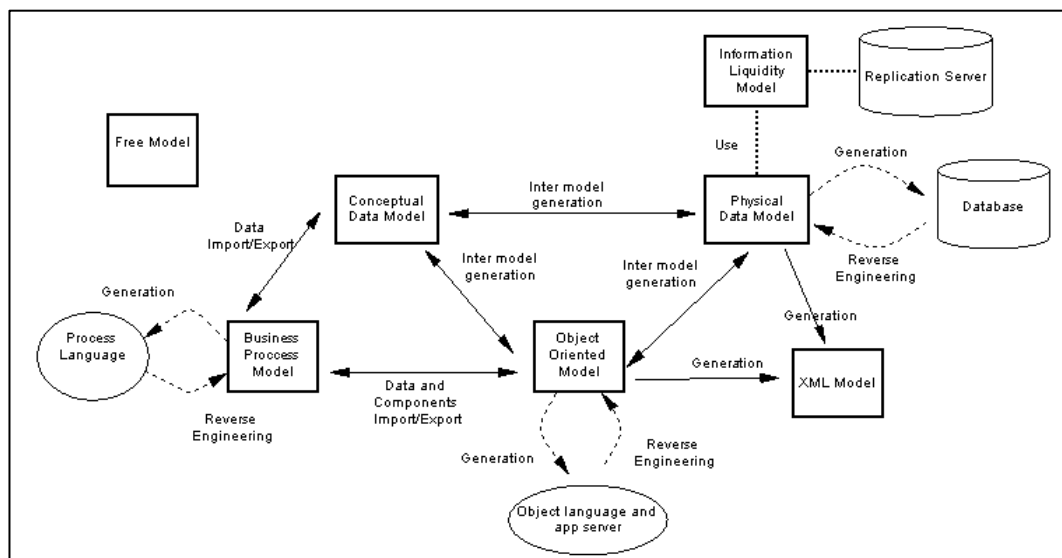
PowerDesigner tidak memaksakan satu metodologi proses rekayasa perangkat lunak tertentu. Setiap perusahaan dapat menerapkan alur kerja sendiri , mendefinisikan tanggung jawab dan peran , menggambarkan alat apa alat yang digunakan, validasi yang diperlukan , dan dokumen apa yang akan diproduksi pada setiap langkah dalam proses. Sebuah tim pengembangan akan terdiri dari *multiple user role* termasuk analist bisnis, analist data dan desainer, database administrator, pengembang, dan penguji, yang masing-masing akan menggunakan kombinasi yang berbeda dari komponen PowerDesigner. Power designer mempunyai beberapa jenis alat yang digunakan untuk membuat pemodelan yang dilakukan oleh: *Business Analysts, Data analysts, Database Administrators, Developers, Team Leaders dan Testers.* Beberapa model dalam power designer yang dapat digunakan antara lain ialah :

1. Model Persyaratan (Requirements Model / RQM) Analist Bisnis dapat mendefinisikan kebutuhan bisnis, yang dapat disempurnakan menjadi persyaratan teknis menggunakan Persyaratan Model ( RQM ). RQM menggambarkan daftar definisi proyek dan menjelaskan fitur apa harus dilaksanakan selama proses pembangunan, dan siapa yang bertanggung jawab. Persyaratan ini melekat pada setiap objek dalam salah satu model lain untuk melacak di mana, dan bagaimana mereka terpenuhi .
2. Proses Model Bisnis (Bussiness process Model / BPM). Analis bisnis dapat menentukan proses bisnis tingkat untuk menggambarkan sistem yang ada dan dapat mensimulasikannya untuk mengurangi waktu dan sumber daya.

BPM menggambarkan proses bisnis dan sebagai alat desain yang mengidentifikasi kebutuhan bisnis, mengatur dalam hirarki, menampilkan proses secara grafis dan menghasilkan komponen dalam bahasa proses seperti BPEL4WS atau Sybase Unwired Orchestrator .

3. Model Data Konseptual (*Conceptual Data Model* atua CDM) merupakan representasi platform-independen dari sistem, memberikan pandangan abstrak struktur basis data. Melalui CDM struktur data real dinormalisasi menjadi relasi satu ke satu, satu kebanyak, banyak-ke-banyak atau hubungan super atau sub-tipe, dan menyediakan pandangan yang jelas dari data bisnis di seluruh sistem Hal ini membuat sistem informasi yang dapat diakses untuk pengguna bisnis, arsitek sistem, dan analis bisnis.
4. Physical Data Model ( PDM ). Setelah struktur data didefinisikan, Database Administrator dapat mengoptimalkan, denormalize dan membuat database menggunakan PDM. PDM merupakan representasi dari database nyata dan obyek terkait yang berjalan pada server melalui informasi lengkap mengenai struktur data fisik, seperti tabel, kolom, referensi, trigger, prosedur, view dan indeks. PDM dapat digunakan untuk menghasilkan semua kode database 50 RDBMSs yang didukung . PDM dapat dibuat dengan reverse engineering dari basis data melalui koneksi database (ODBC). PDM dan CDM dapat memastikan bahwa implementasi akhir sama persis dengan persyaratan sistem yang sebenarnya.
5. Model data logis (Logical Data Model / LDM ) merupakan versi RDBMS khusus yang independen dariPDM.Model ini digunakan sebagai jembatan antara CDM dan PDM. Melalui LDM memungkinkan sistem analist untukk menyelesaikan relasi banyak - ke-banyak, hubungan super atau sub tipe , denormalisasi struktur basis data dan menentukan indeks tanpa menggunakan RDBMS tertentu .
6. Model Likuiditas Informasi (ILM). Jika pengguna bertanggung jawab untuk replikasi database maka dapat menggunakan Model Likuiditas Informasi (ILM). Model ini menyediakan representasi global replikasi informasi dari database sumber ke satu atau beberapa remote database .
7. Object-Oriented Model (OOM) digunakan untuk membuat pemodelan berbasis obyek yang menggunakan diagram standar UML dan notasi yang mewakili obyek-obyek dan interaksinya. Model ini dapat digunakan untuk

menghasilkan kode program dalam bahasa Java,. NET dan bahasa pemrograman lainnya.

8. XML Model (XSM) digunakan untuk membuat pemodelan grafis untuk struktur yang komplek dari file XML tampilan diagram pohon dalam model ini memberikan pandangan global dan skematis dari semua elemen dokumen, dan jenis model yang digunakan untuk menghasilkan DTD, dan XSDs langsung dari PDM atau OOM.
9. Model bebas (Free model/FEM). Model ini digunakan untuk membuat diagram yang menjelaskan arsitektur sistem dan aplikasi, skenario use case aplikasi, diagram alur, dan model grafis lainnya.

Gambar48. Iinteraksi model dalam power designer

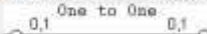
Gambar diatas menjelaskan bagaimana berbagai model yang disediakan oleh tools power designer dapat berinteraksi selama proses perancangan, pemeliharaan dan pendistribusian sistem.

### 3) Conceptual Data Model (CDM)

Conceptual Data Model ( CDM ) merupakan struktur logis dari keseluruhan database, yang terpisah dari perangkat lunak dan struktur penyimpanan data. CDM memberikan representasi formal dari data yang diperlukan untuk menjalankan suatu perusahaan atau kegiatan usaha dan meliputi objek data atau entitas dalam database logis atau konseptual. Dalam merancang sebuah

database, proses desain biasanya dimulai pada tingkat konseptual, di mana pengguna tidak perlu mempertimbangkan rincian implementasi fisik yang sebenarnya. CDM memungkinkan pengguna untuk :

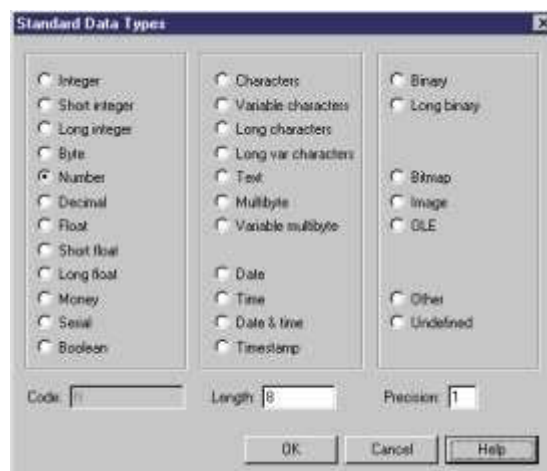
- ✓ Mewakili penglolaan data dalam format grafis untuk membuat Entity Relationship Diagram ( ERD )
- ✓ Memverifikasi keabsahan desain data
- ✓ Menghasilkan Physical Data Model (PDM), yang akan menentukan implementasi fisik database
- ✓ Dapat menghasilkan Model Object Oriented (OOM), yang akan menentukan representasi objek CDM menggunakan standar UML
- ✓ Menghasilkan CDM lain, yang akan membuat versi model lain untuk mewakili tahap desain yang berbeda.

CDM menjelaskan diagram relasi entitas untuk level konspetual. Entitas adalah representasi obyek atau data dari dunianya. Entitas dapat berupa nama benda, nama orang, nama tempat, atau kejadian. Gambar dibawah ini menjelaskan berbagai ragam jenis simbol atau notasi beserta diskripsinya dalam power designer yang dapat digunakan untuk membuat diagram CDM.

| Object | Tool | Symbol | Description |
|---|---|---|---|
| Domain | [none] | [none] | Set of values for which a data item is valid. See Domains (CDM). |
| Data Item | [none] | [none] | Elementary piece of information. See Data Items (CDM). |
| Entity | | Entity | Person, place, thing, or concept that has characteristics of interest to the enterprise and about which you want to store information. See Entities (CDM). |
| Entity Attribute | [none] | [none] | Elementary piece of information attached to an entity. See Attributes (CDM). |
| Identifier | [none] | [none] | Entity attribute, or a combination of entity attributes, whose values uniquely identify each occurrence of the entity. See Identifiers (CDM). |
| Relationship | | | Named connection or relation between entities (Entity Relationship (ER) modeling methodology). See Relationships (CDM). |
| Inheritance | | | Special relationship that defines an entity as a special case of a more general entity. See Inheritances (CDM). |
| Association | | Association | Named connection or association between entities (Merise modeling methodology). See Associations and Association Links (CDM). |
| Association Link | | | Link that connects an association to an entity and on which you define the cardinality an entity has relative to another. See Associations and Association Links (CDM). |

Gambar 49. Ragam jenis notasi atau simbol CDM

Sementara itu gambar dibawah ini menjelaskan berbagai ragam pilihan jenis atau tipe relasi antara dua Entitas, yaitu *one to one, one to many, many to one dan many to many*. Relasi *many to many* juga dapat dibentuk menggunakan notasi assosiation yang menjelaskan relasi antara dua entitas atau lebih (N-nary)

| Option | Description |
|---|---|
| Entity / Relationships | [Default – used throughout this manual] Entity/relationship notation connects entities with links representing one of four relationships between them. These relationships have properties that apply to both entities involved in the relationship<br>One to One (0,1 – 0,1)<br>One to Many (0,1 – 0,n)<br>Many to One (0,n – 0,1)<br>Many to Many (0,n – 0,n) |
| Merise | Merise notation uses associations instead of relationships<br>0,n – Assc_1 – 0,n |
| E/R + Merise | Both entity/relationship and Merise are used in the same model |
| IDEF1X | Data modeling notation for relationships and entities. In this notation, each set of relationship symbols describes a combination of the optionality and cardinality of the entity next to it<br>Non mandatory parent<br>Mandatory parent<br>Dependent<br>0,n<br>1,1<br>0,1<br>1,n |

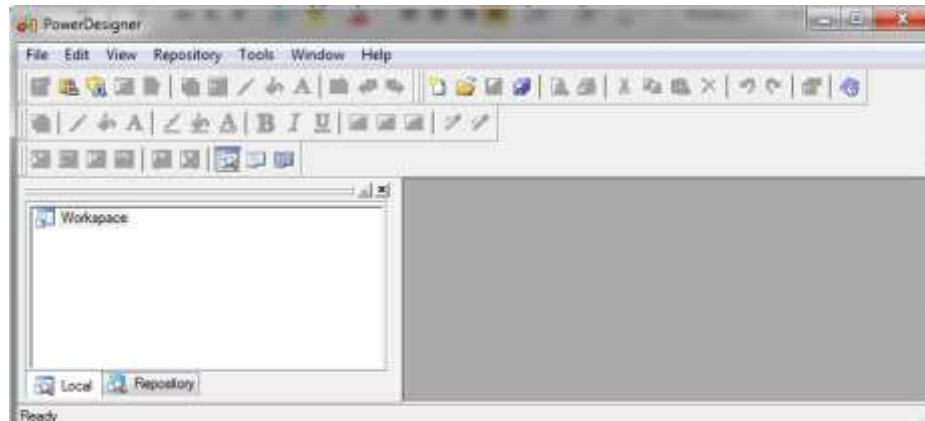
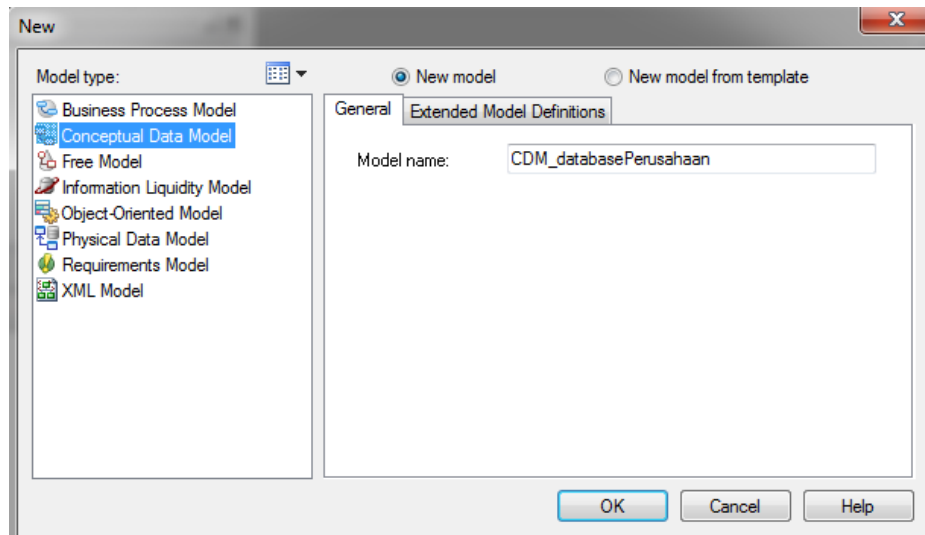
Gambar 50. berbagai ragam jenis relasi entitas dan notasinya

Gambar 51. Ragam standar type data untuk atribut dari entitas

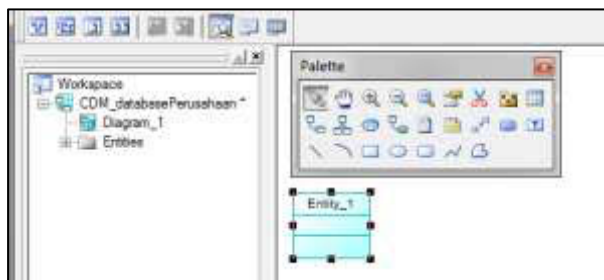
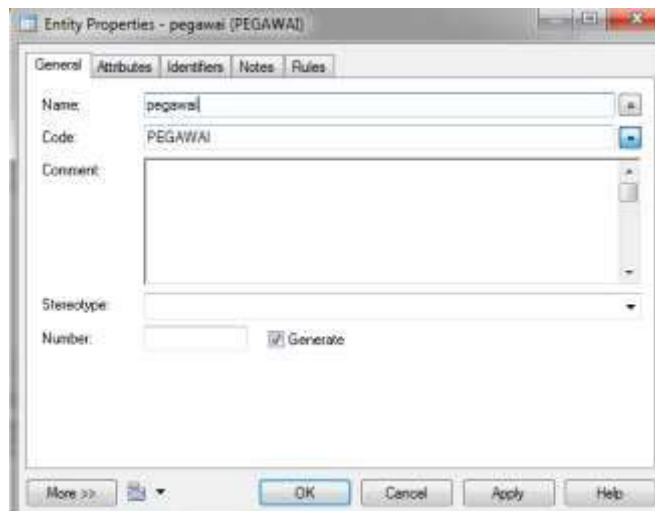
| Conceptual data type | DBMS-specific physical data type | Content | Length |
|---|---|---|---|
| Integer | int / INTEGER | 32-bit integer | — |
| Short Integer | smallint / SMALLINT | 16-bit integer | — |
| Long Integer | int / INTEGER | 32-bit integer | — |
| Byte | tinyint / SMALLINT | 256 values | — |
| Number | numeric / NUMBER | Numbers with a fixed decimal point | Fixed |
| Decimal | decimal / NUMBER | Numbers with a fixed decimal point | Fixed |
| Float | float / FLOAT | 32-bit floating point numbers | Fixed |
| Short Float | real / FLOAT | Less than 32-bit point decimal number | • — |
| Long Float | double precision / BINARY DOUBLE | 64-bit floating point numbers | — |
| Money | money / NUMBER | Numbers with a fixed decimal point | Fixed |
| Serial | numeric / NUMBER | Automatically incremented numbers | Fixed |
| Boolean | bit / SMALLINT | Two opposing values (true/false; yes/no; 1/0) | — |
| **Conceptual data type** | **DBMS-specific physical data type** | **Content** | **Length** |
| Characters | char / CHAR | Character strings | Fixed |
| Variable Characters | varchar / VARCHAR2 | Character strings | Maximum |
| Long Characters | varchar / CLOB | Character strings | Maximum |
| Long Var Characters | text / CLOB | Character strings | Maximum |
| Text | text / CLOB | Character strings | Maximum |
| Multibyte | nchar / NCHAR | Multibyte character strings | Fixed |
| Variable Multibyte | nvarchar / NVARCHAR2 | Multibyte character strings | Maximum |
| **Conceptual data type** | **DBMS-specific physical data type** | **Content** | **Length** |
| Date | date / DATE | Day, month, year | |
| Time | time / DATE | Hour, minute, and second | |
| Date & Time | datetime / DATE | Date and time | |
| Timestamp | timestamp / TIMESTAMP | System date and time | |
| **Conceptual data type** | **DBMS-specific physical data type** | **Content** | **Length** |
| Binary | binary / RAW | Binary strings | Maximum |
| Long Binary | image / BLOB | Binary strings | Maximum |
| Bitmap | image / BLOB | Images in bitmap format (BMP) | Maximum |
| Image | image / BLOB | Images | Maximum |
| OLE | image / BLOB | OLE links | Maximum |
| Other | — | User-defined data type | — |
| Undefined | undefined | Not yet defined data type | — |

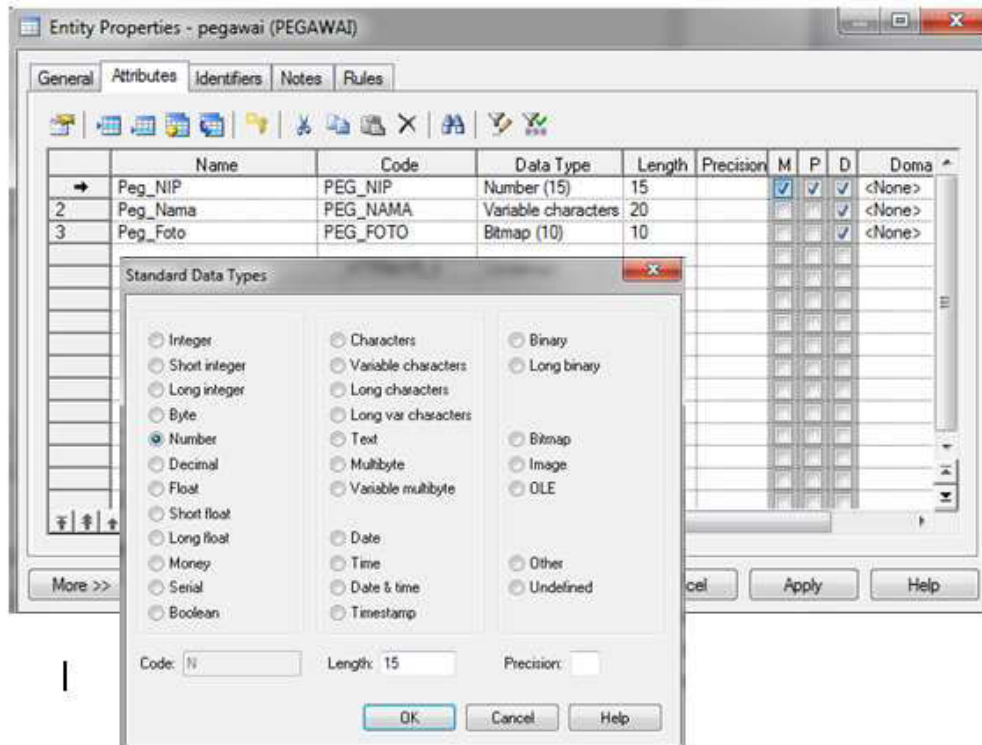
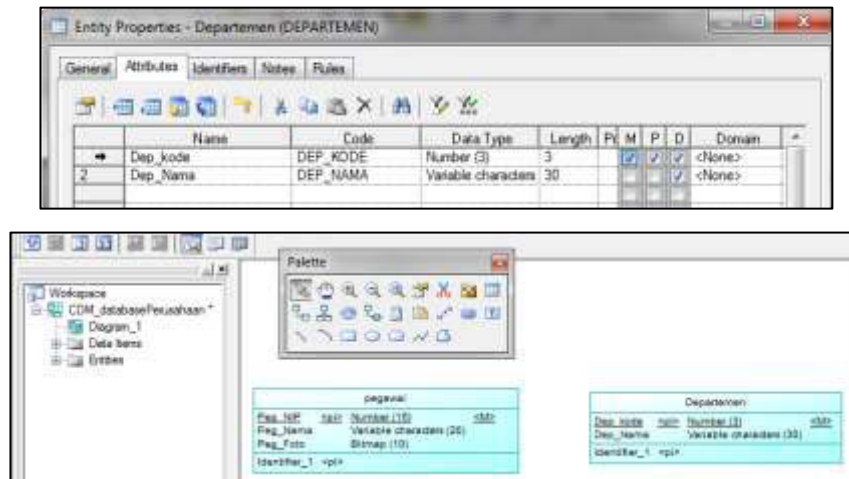
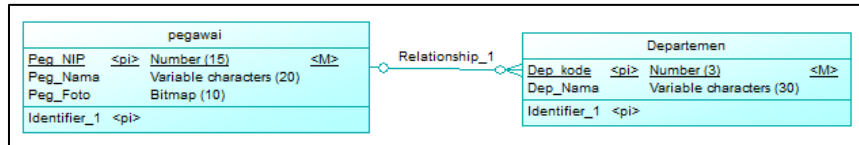
Gambar 52. Ragam type data dan jangkauan variabel dalam CDM

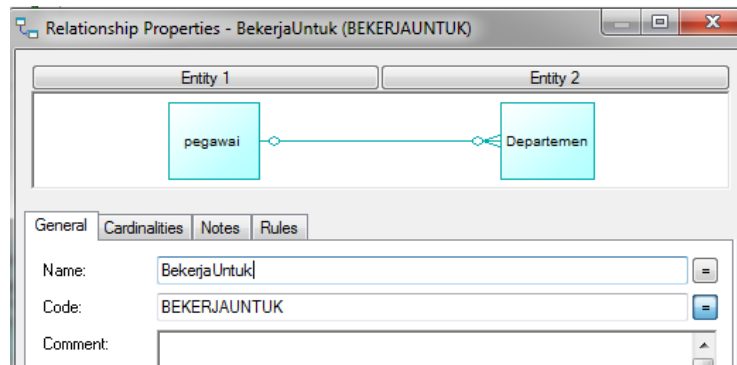
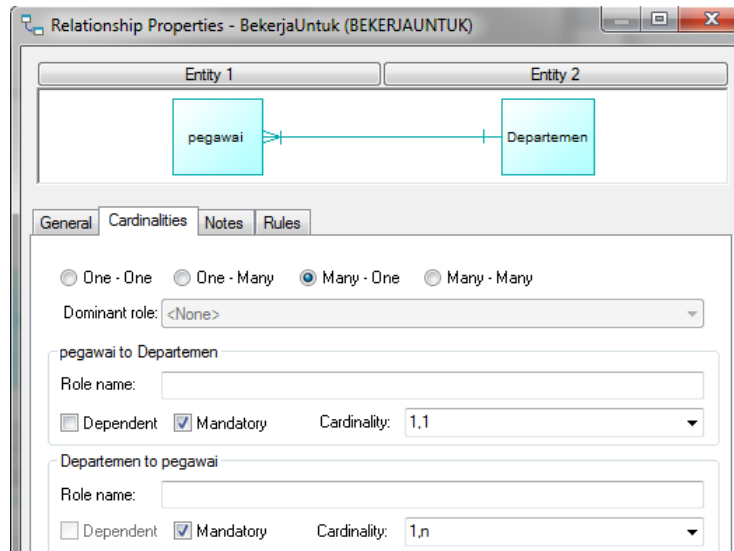
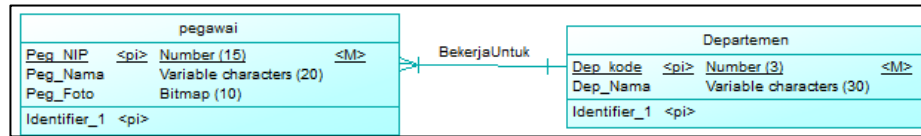
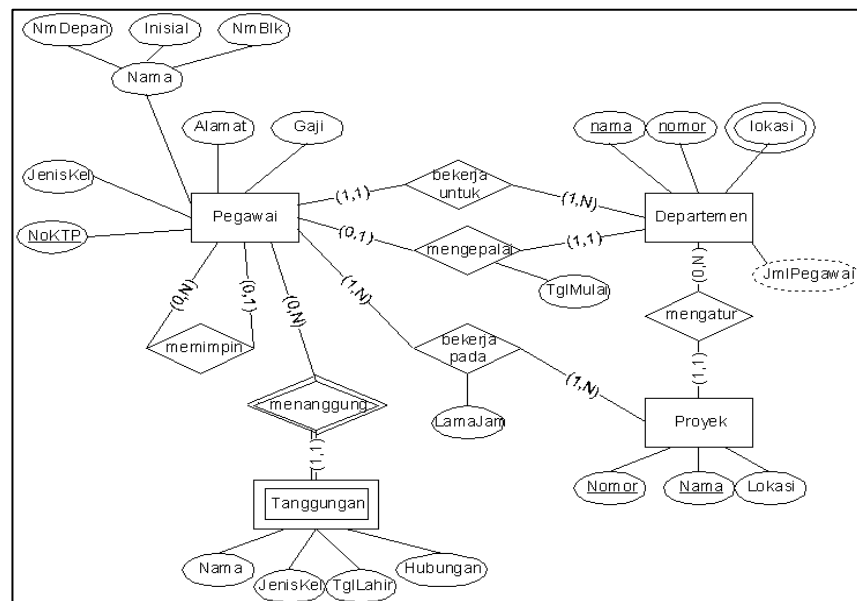
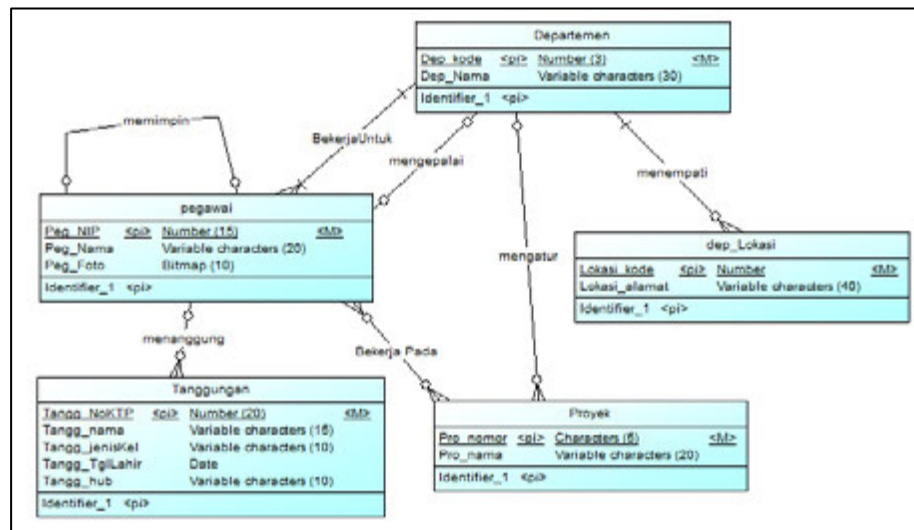
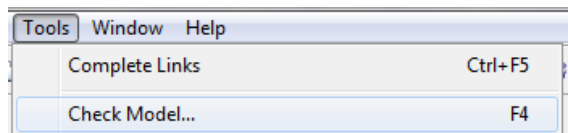
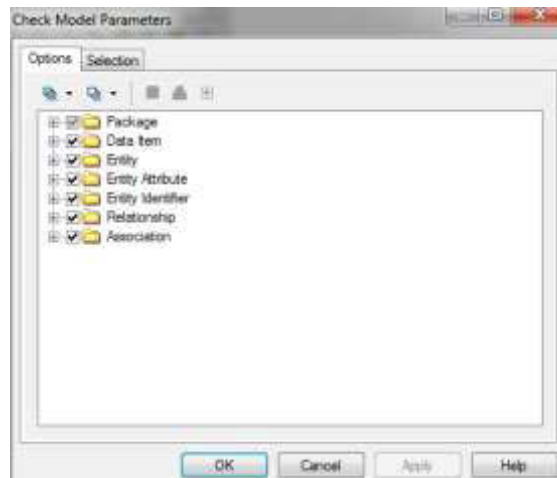
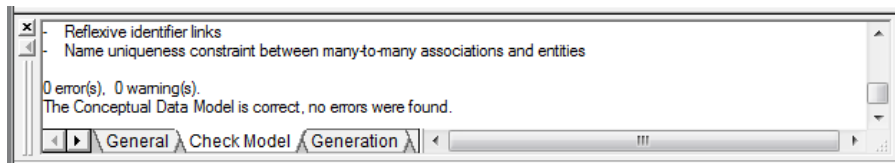
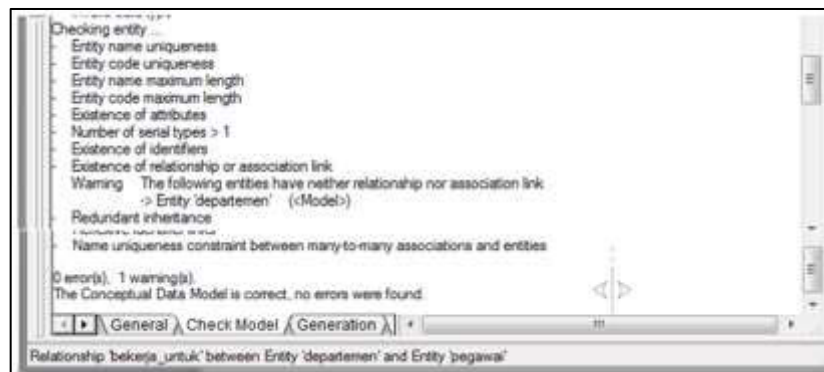
### 4) Membuat diagram CDM

Setelah melakukan instalasi program power designer pengguna dapat membuat diagram CDM baru dari awal, atau dengan meng-*impor* dari diagram Analyst Model Process (.PAM) atau model Erwin (.ERX). Diagram CDM juga dapat dihasilkan dari proses *reverse engenering* atau *generate* dari diagram PDM, atau OOM. Beberapa langkah yang dilakukan untuk membuat diagram CDM yaitu sebagai berikut:

1. Instal aplikasi sybase power Designer
2. Buka aplikasi power designer

3. Dari menu pilih file → new atau menekan ctrl + N

4. Pilih Conceptual Data Model dan klik *radio buttom new model*
   - ✓ New model : digunakan untuk membuat standar CDM baru
   - ✓ New model from template: digunakan untuk membuat CDM dari model Template. Model Template adalah seperangkat pilihan model, preferensi tampilan, ekstensi, atau penyimpanan model obyek dalam folder Template. Template ini digunakan ketika pengguna harus menggunakan kembali preferensi dan pilihan dalam beberapa model
5. Ketik nama model dalam kotak nama Model misalnya: CDM_dataBasePerusahaan
6. Jika pengguna ingin melampirkan atau menambahkan satu atau lebih model maka klik tab extended model definition
7. Klik OK untuk membuat CDM baru dalam Workspace.

8. Tambahkan obyek entitas dengan drag and drop icon pada tool palete

9. Klik dua kali pada obyek Entity_1 untuk merubah propertis dari entitas

10. Ketik nama entitas pada *tex box name* misalnya : pegawai

11. Pada baris code Klik icon , maka nama code secara otomatis sama dengan nama entitas dan dicetak menggunakan huruf capital
12. Klik tab attributes untuk menambahkan data atribute dari entitas pegawai

13. Tambahkan nama attribut pada kolom Name: Peg_NIP, tambahkan data type : Number dan panjang variabel (Length) :15, contreng chek Box kolom P yang menandakan *primary key* dan secara otomatis chek box kolom M (mandatory) dan Check box kolom D (Domain) akan tercontreng. Klik ok
14. Tambahkan pula untuk attribute-atribute lainnya. Klik apply dan ok
15. Ulangi langkah h untuk menambahkan entitas departemen dengan propertis sebagai berikut

16. Tambahkan obyek relationship dengan klik icon  arahkan pointer ke entitas pegawai, klik kiri dan tahan kemudian gerakkan mouse ke entitas departemen kemudian lepas klik kiri.


17. Klik dua kali pada obyek relationship_1 untuk mengubah propertis relasi

18. Ketik nama relasinya pada tex *box name*. Misal: bekerja untuk
19. Klik tab cardinalitas untuk mengubah propertis kardinalitas

20. Klik *radio buttom* pada *many to one*. Untuk relasi pegawai to departemen pilih cardinality 1,1 dan relasi departemen to pegawai cardinality 1,nd an klik ok

21. Relasi antara entitas pegawai dan departemen sudah terbentuk. Gambar diatas menjelaskkan bahwa : setiap pegawai harus bekerja pada satu depertemen dan setiap departemen terdiri dari banyak pegawai.
22. Dengan cara yang sama bisa dilakukan untuk entitas dan relasi lainnya sesuai dengan ERD sistem perusahaan sebagaimana telah dibahas dalam kegiatan belajar 4.

23. Atur dan perbaiki dokumen CDM yang telah dibuat pada langkah-langkah diatas dan sesuikan nama atribute dan entitas seperti gambar ERD diatas.

24. Keterangan :
    - ✓ atribute yang ditambahkan dalam CDM adalah seluruh simple atribute
    - ✓ untuk atribute multivalue pada ERD, buatlah entitas baru pada CDM dengan nama entitas seperti nama atribute multivalue (misal: dep_lokasi) hubungkan dengan relasi *one to many* dengan *many* disisi entitas Dep_lokasi
    - ✓ untuk memudahkan dalam membedakan suatu atribut merupakan atribut entitas yang mana, maka penulisan atribute ditambahkan nama entitasnya, misal nama atribut : peg_nama (atribute nama dari entitas pegawai, dep_nama (atribute nama dari entitas departemen), tanggung_nama (atribute nama dari entitas tanggungan)
25. setelah semua entitas dan atribut sudah ditambahkan dalam CDM langkah selanjutnya cek model. Pilih menu tools → check model atau (F4)

26. Contreng semua folder kemudian klik OK. Jika jumlah error = 0 dan warning = 0 maka pembuatan CDM sudah benar

Jika terjadi kesalahan dan peringatan maka program akan dapat menampilkan jenis kesalahan dan peringatannya.

## c. Rangkuman

PowerDesigner adalah software tools dengan pendekatan model driven berbasis grafis yang digunakan untuk untuk menyelaraskan bisnis dan Teknologi informasi. Tools ini merupakan *enterprise modelling* yang membantu mengimplementasikan Enterprise Architecture dan membawa lingkungan manajemen meta-data yang kuat untuk siklus hidup pengembangan aplikasi. Berbagai jenis ragam model yang dapat dibuat menggunakan power designer adalah 1) Model Persyaratan (Requirements Model / RQM). 2) Proses Model

Bisnis (Bussiness process Model / BPM). 3) Model Data Konseptual (*Conceptual Data Model* / CDM). 4) Physical Data Model ( PDM ). 5) Model data logis (Logical Data Model / LDM ). 6) Model Likuiditas Informasi (ILM). 7) Object-Oriented Model (OOM). 8) XML Model (XSM). 9) Model bebas (Free model/FEM).

Conceptual Data Model ( CDM ) merupakan struktur logis dari keseluruhan database, yang terpisah dari perangkat lunak dan struktur penyimpanan data. CDM memberikan representasi formal dari data yang diperlukan untuk menjalankan suatu perusahaan atau kegiatan usaha dan meliputi objek data entitas, relasi antar entias dalam database logis atau konseptual.

**d. Tugas : Membuat Conceptual data Model (CDM)**

Dalam kegiatan ini peserta didik akan melakukan praktikum secara berkelompok, satu kelompok terdiri dari dua sampai tiga orang. Peserta didik akan merancang pemodelan data level konseptual (conceptual data model). Bacalah seluruh langkah pengamatan dibawah ini kemudian lakukan dengan cermat dan teliti dengan perangkat yang telah disediakan.

1. Bentuk kelompok diskusi setiap kelompok terdiri dari tiga orang.
2. Instal aplikasi program sybase powerdesigner 12.5 atau yang terbaru. Amati setiap tampilan atau perubahan dilayar monitor dan catat hasilnya.
3. Berdasarkan uraian dari materi pembelajaran buatlah diagram CDM sistem database perusahaan. Tambahkan obyek entitas, relasi antar entitas. Atur propertis entitas dan relasi sesuai dengan ERD database perusahaan. Catat hasilnya.
4. Kumpulkan data-data setiap langkah dan analisis data tersebut menggunakan analisis diskriptif.
5. Komunikasikan hasilnya dalam kelompok dan buatlah kesimpulan.
6. Presentasikan hasil diskusi bersama-sama dengan kelompok lainnya dan guru pembimbing.

**e. Test Formatif.**

Dalam test ini setiap peserta didik membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

1. Sebutkan dan jelaskan secara singkat ragam jenis model diagram yang dapat dibuat menggunakan tools power designer ?
2. Sebutkan dan Jelaskan berbagai ragam jenis relasi, cardinality dan berikan contoh relasi beserta cardinality relasi tersebut.?

**f. Lembar Jawaban Test Formatif (LJ).**

**LJ- 01 : Ragam jenis model diagram dalam tools power designer.**

..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................
..........................................................................................

**Kegiatan belajar 12: Alat bantu pemodelan data fisik**

**a. Tujuan Pembelajaran.**

Setelah mengikuti kegiatan belajar10 ini diharapkan peserta didik dapat:

- ✓ Memahami konsep pemodelan data fisik
- ✓ Mengoperasikan tools pemodelan data fisik menggunakan power designer

**b. Uraian materi.**

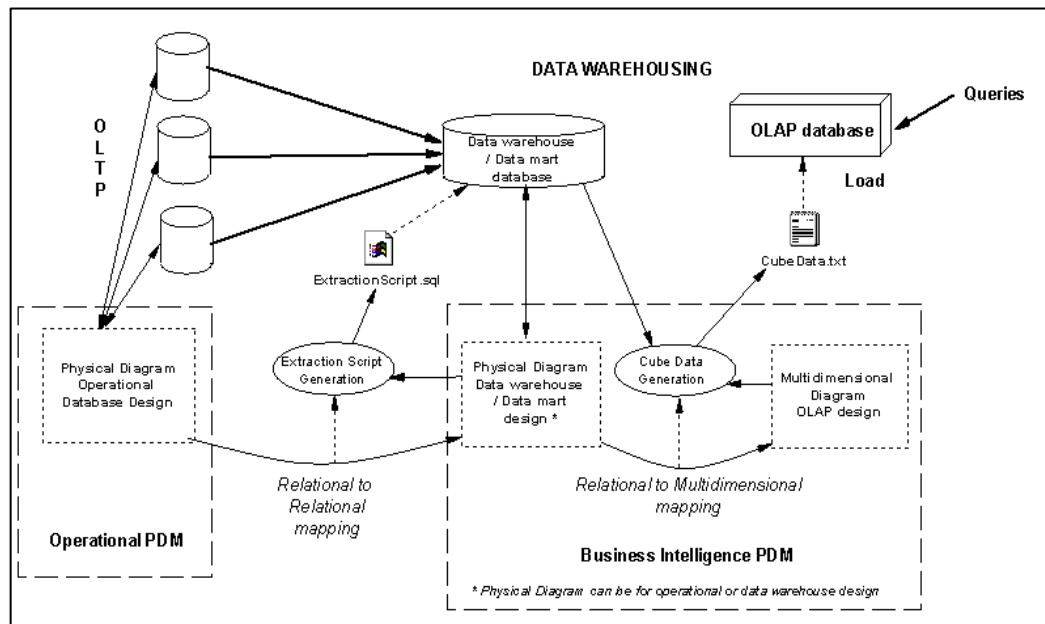
**1) Physical Data Model (PDM)**

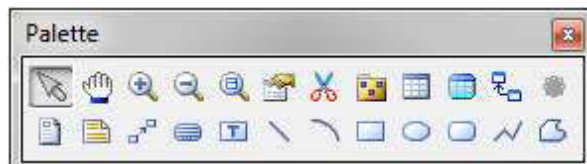
PDM adalah merupakan suatu model implementasi phisik dari database. Dengan PDM, pengguna dapat mempertimbangkan secara detil tentang implementasi phisik nyata. PDM memasukkan kedalam laporan perangkat lunak atau penyimpanan data struktur. Pengguna dapat memodifikasi PDM dengan menyesuaikan batasan phisik (physical constrain) atau hasil rancangan. PDM mempunyai beberapa peran sebagai berikut :

- Menghadirkan organisasi phisik data di (dalam) suatu format grafis
- Menghasilkan catatan yang digunakan untuk memodifikasi dan pembuatan database
- Menggambarkan batasan (constrain) dan referensi integritas
- Menghasilkan extended atribut
- Merekayasa balik database yang ada
- Memperbaharui suatu CDM

Ada beberapa jalan untuk membuat suatu PDM:

- Membuat suatu dokumen PDM sejak dari awal mula
- Membuat PDM dengan mengkonversi (generate) dari dokumen CDM
- Mengkonversi dari suatu database ke dalam suatu PDM

| Database | Diagram |
|---|---|
| Operational | Diagram fisik untuk menentukan pelaksanaan fisik database |
| Date warehouse or Data mart | Diagram fisik untuk menyimpan data bisnis |
| OLAP | Diagram multidimensi untuk mendefinisikan query mungkin untuk melakukan pada data operasional |

### 2) Pengenalan obyek pada toolsbocs PDM

Gambar dibawah ini merupakan beberapa icon atau tools yang dapat digunakan untuk membuat diagram model fisik basis data menggunakan alat bantu power designer.

Gambar Pallete pada toolbox

Tabel dibawah ini menjelaskan beberapa ragam obyek, icon atau tools, simbol dan diskripsinya untuk pemodelan data fisik pada alat bantu power designer.

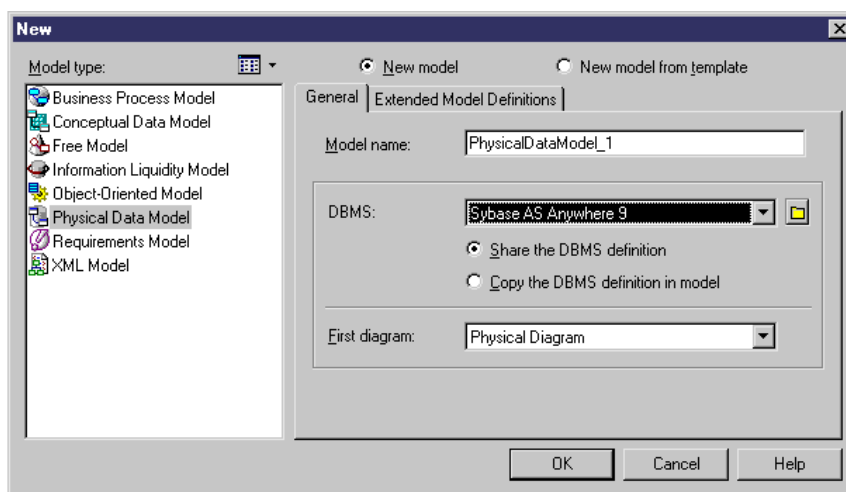
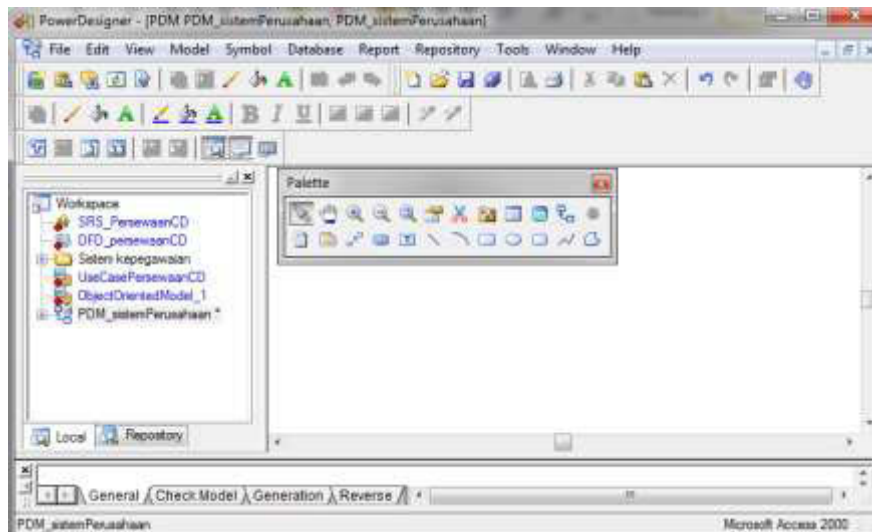
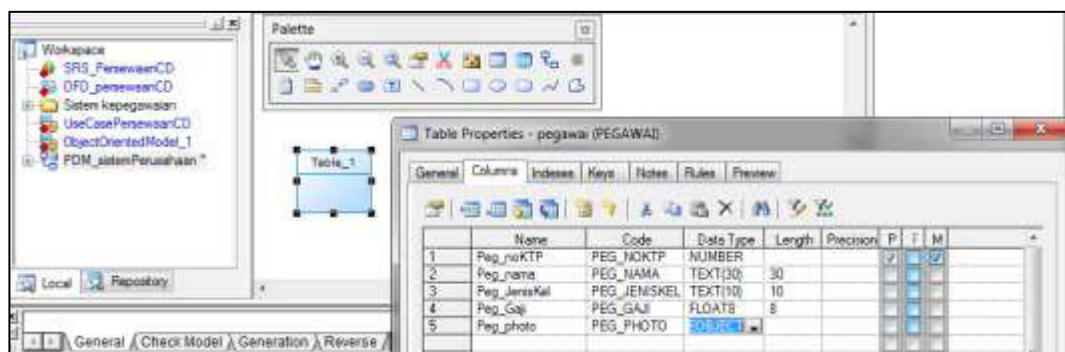
| Obyek | Alat | Simbol | Deskripsi |
|---|---|---|---|
| **Table** | | Table | Koleksi baris (record) yang telah dikaitkan dengan kolom (field). Lihat Tables (PDM). |
| **Column** | [none] | [none] | Struktur data yang berisi item data individu dalam baris (record), model ekivalen bidang database. Lihat Columns (PDM). |
| **Primary key** | [none] | [none] | Kolom yang meiliki nilai unik mengidentifikasi setiap baris dalam sebuah tabel, dan ditetapkan sebagai identifier utama dari setiap baris dalam tabel. lihat Keys (PDM). |

| Obyek | Alat | Simbol | Deskripsi |
|---|---|---|---|
| **Alternate key** | [none] | [none] | Kolom atau kolom yang nilai unik mengidentifikasi setiap baris dalam sebuah tabel, dan yang bukan merupakan primary key. lihat Keys (PDM). |
| **Foreign key** | [none] | [none] | Kolom yang nilainya bergantung pada dan bermigrasi dari primary key atau alternatif di meja lain. lihat Keys (PDM). |
| **Index** | [none] | [none] | Struktur data yang terkait dengan satu atau lebih kolom dalam sebuah tabel, di mana nilai-nilai kolom yang diperintahkan sedemikian rupa untuk mempercepat akses ke data. lihat Indexes (PDM). |
| **Default** | [none] | [none] | [DBMS tertentu] Sebuah nilai default untuk kolom. lihat Defaults (PDM). |
| **Domain** | [none] | [none] | Mendefinisikan nilai yang valid untuk kolom. lihat Domains (PDM). |
| **Sequence** | [none] | [none] | [DBMS tertentu] Mendefinisikan bentuk incrementation untuk kolom. lihat Sequences (PDM). |
| **Abstract data type** | [none] | [none] | [DBMS tertentu] User-didefinisikan tipe data. lihat Abstract Data Types (PDM). |
| **Reference** | | | Link antara primer atau kunci alternatif dalam tabel induk, dan kunci asing dari tabel anak. Tergantung pada sifat-sifatnya yang dipilih, referensi juga dapat menghubungkan kolom yang independen terhadap kolom kunci primer atau alternatif. lihat References (PDM). |
| **View** | | View | Struktur data yang dihasilkan dari query SQL dan yang dibangun dari data dalam satu atau lebih tabel. lihat Views (PDM). |
| **View Reference** | | View | Hubungan antara meja dan pandangan. lihat View References (PDM). |
| **Trigger** | [none] | [none] | Segmen kode SQL yang berhubungan dengan tabel atau pandangan. Lihat "Pemicu Sekilas di Gedung Pemicu dan Prosedur bab. |
| **Procedure** | | Proc | Koleksi dikompilasi pernyataan SQL disimpan di bawah nama dalam database dan diproses sebagai satu unit. Lihat "Stored Prosedur dan Fungsi" di Gedung Pemicu dan Prosedur bab. |
| **Database** | [none] | [none] | Database yang PDM merupakan representasi sebuah. Lihat "Membuat Database" dalam bab Physical Data Model Dasar. |
| **Storage** | [none] | [none] | Sebuah partisi pada perangkat penyimpanan. Lihat "Tabel ruang dan Storages" dalam Membangkitkan Database dari bab PDM. |
| **Tablespace** | [none] | [none] | Sebuah partisi dalam database. Lihat "Tabel ruang dan Storages" dalam Membangkitkan Database dari bab PDM. |

| Obyek | Alat | Simbol | Deskripsi |
|---|---|---|---|
| **User** | [none] | [none] | Seseorang yang dapat log in atau koneksi ke database. Lihat "Pengguna (PDM)" di Gedung bab Struktur Database Access. |
| **Role** | [none] | [none] | A predefined user profile. See "Roles (PDM)" in the Building a Database Access Structure chapter. |
| **Group** | [none] | [none] | Mendefinisikan hak dan izin untuk satu set pengguna. Lihat "Grup (PDM)" di Gedung bab Struktur Database Access. |
| **Synonym** | [none] | [none] | Nama alternatif untuk berbagai jenis benda. Lihat "Sinonim (PDM)" di Gedung bab Struktur Database Access. |
| **Web service** | [none] | [none] | Koleksi pernyataan SQL yang disimpan dalam database untuk mengambil data relasional di HTML, XML, WSDL atau format teks biasa, melalui HTTP atau permintaan SOAP. Lihat "Web Services (PDM)" di Web Services Building bab. |
| **Web operation** | [none] | [none] | Sub-objek layanan Web yang berisi pernyataan SQL dan menampilkan parameter Web dan kolom hasil. Lihat "Web Service Operations (PDM)" dalam bab Web Services Building. |

**3) Membuat diagram PDM**

PDM dapat dibuat dengan dua cara yaitu: 1) membuat dokumen PDM baru. 2) mengkonversi dari dokumen CDM. Uraian dibawah ini menjelaskan langkah-langkah pembuatan diagram fisik basis data (PDM) menggunakan alat bantu software power designer

1. Pilih File → New untuk menampilkan kotak dialog New.

2. Pilih Physical Data Model pada daftar jenis Model.
3. Pilih salah satu tombol option sebagai berikut:
   - *New Model* - Membuat standar model baru.
   - *New Model from* - Membuat model dari model Template. Sebuah model Template adalah seperangkat pilihan model, preferensi tampilan, ekstensi, atau benda-benda yang disimpan dalam model terletak di folder Template. Pengguna dapat menggunakan model template ketika harus menggunakan kembali preferensi dan pilihan dalam beberapa model.
4. Ketik nama model dalam kotak nama Model. Kode dari model, yang dapat digunakan untuk skrip atau generasi kode, berasal dari nama ini sesuai dengan model konvensi penamaan.Select a DBMS from the list.
5. DBMS didefinisikan dalam file XML khusus (dengan ekstensi. Xdb), yang disediakan sebagai bagian dari instalasi PowerDesigner di "\ Resource Files \ DBMS" direktori, dan berisi semua sintaks dan spesifikasi untuk setiap target DBMS. Pilih salah satu dengan mengklik tombol option:
   - Bagi definisi DBMS - menggunakan file DBMS asli dalam "Resource Files \ DBMS" direktori. Setiap perubahan yang dibuat ke DBMS dibagi oleh semua PDM terkait.
   - Salin definisi DBMS dalam model - membuat salinan file DBMS asli dalam "Resource Files \ DBMS" direktori. DBMS saat ini adalah independen dari DBMS asli, sehingga modifikasi yang dilakukan DBMS dalam direktori DBMS tidak tersedia untuk PDM.
6. Pilih jenis diagram pertama dalam daftar Diagram Pertama. Jenis pertama dari diagram yang dipilih tetap dalam memori, dan merupakan default untuk waktu berikutnya ketika Anda membuat PDM baru.
7. Pengguna dapat membuat beberapa diagram yang dibutuhkan dalam satu dokumen PDM yang sama. Diagram tersebut diurutkan menurut abjad pada jendela bro*wser.*
8. Jika pengguna ingin melampirkan satu atau lebih definisi model untuk melengkapi DBMS yang dipilih, klik tab *Extended Model Definition*.
9. Kemudian Klik OK untuk membuat PDM baru dalam Workspace tersebut.

10. Tambahkan object tabel dengan drag icon tabel dari jendela palette ke lember kerja.

11. Edit propertis dengan klik dua kali pada object tabel1. Klik tab general dan ketik : pegawai pada baris name.
12. Klik tab columns dan tambahkan atribute-atribute tabelnya beserta tipe datanya kemudian klik oke
13. Ulangi langkah 10 untuk tabel departemen dan tabel lainnya

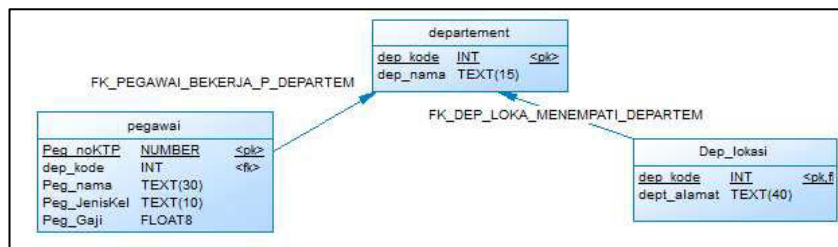
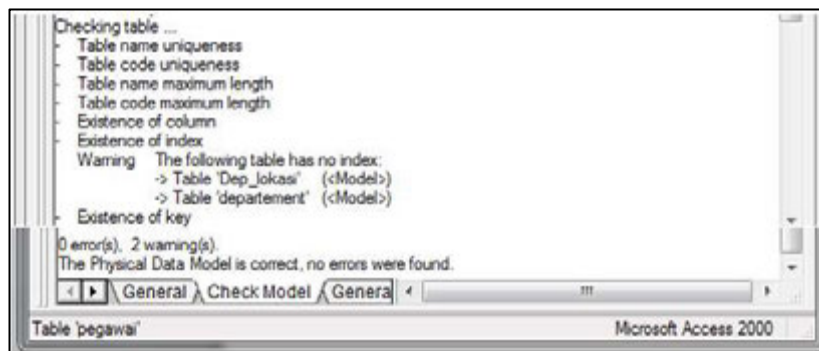
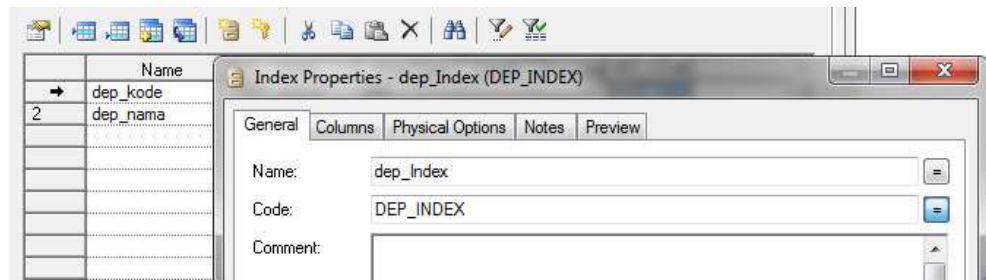
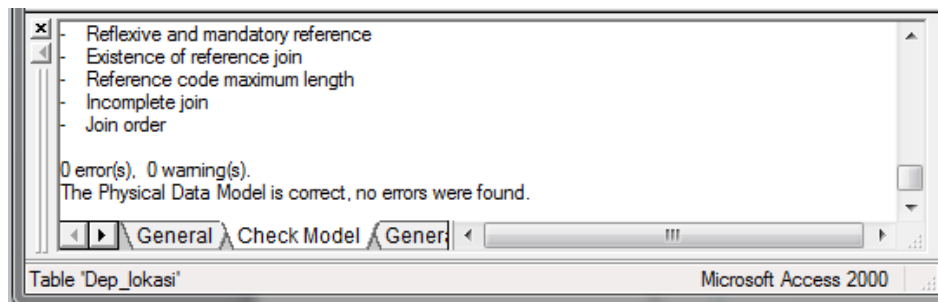
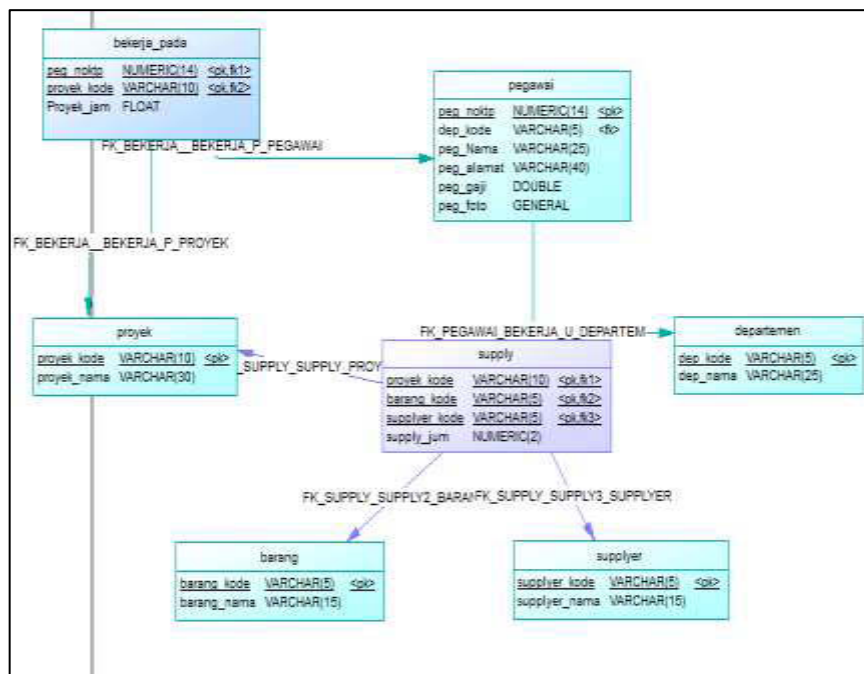
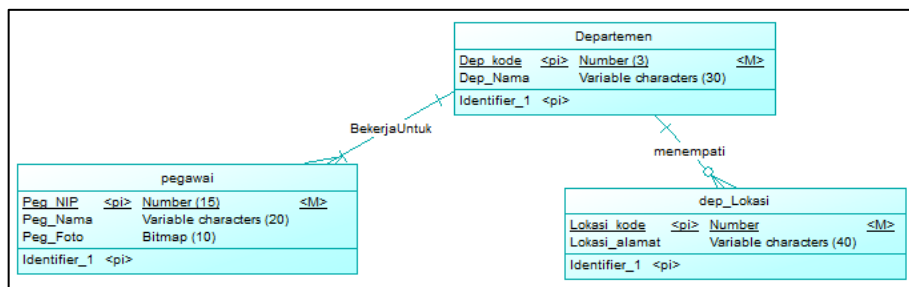
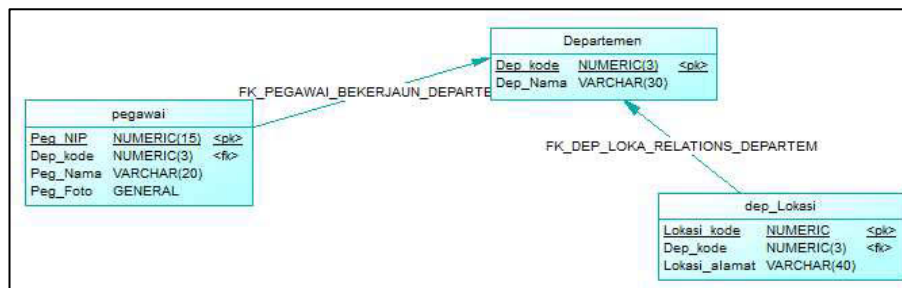
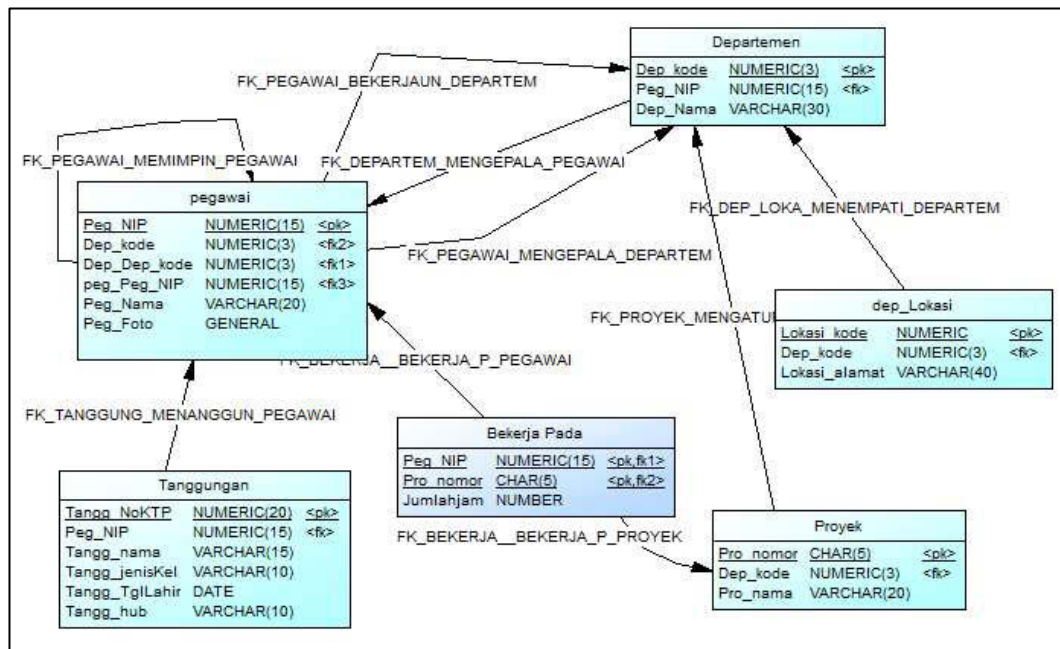
14. Tambahkan object reference dengan mengklik icon. Arahkan kursor ke tabel pegawai, klik kiri mouse dan tahan kemudian gerakkan mouse ke tabel departemen kemudian lepaskan mouse. Maka tabel pegawai dan akan berelasi dengan tabel departemen dengan tabel departemen sebagai child dan departemen sebagi parent. Secara otomatis atribute dep_kode (PK) tabel departemen akan ditambahkan ke tabel peawai sebagi atribute forign key (FK).
15. Edit nama relasi dengan mengklik obyek reference dan ketik nama relasi : bekerja pada.
16. Ulangi langkah 14 untuk menghubungkan tabel repartemen dengan dep_lokasi.

17. Cek model diagram dengan menekan menu tools check model atau tombol F4

18. Peringatan yang muncul dari cek model adalah untuk ketiga tabel diatas belum ada tabel indeksnya
19. Untuk menambahkan tabel indeks, double klik tabel departemen, klik tab columns dan tambahkan indeks dengan mengklik icon ketik nama indeks misal dep_index kemudian klik oke

20. Ulangi langkah 19 untuk menambahkan index pada tabel-tabel lainnya.
21. Cek model dengan mengklik tombol F4

Gambar dibawah ini menjelaskan salah satu contoh diagram fisik basis data (PDM) menggunakan alat bantu software power designer.

**4) Membuat diagram PDM dari diagram CDM**

Diagram PDM juga dapat dibuat dari diagram CDM dengan menggunakan fungsi *generate physical data Model*. Langkah-langkah yang dilakukan untuk membuat diagram PDM dari diagram CDM adalah sebagai berikut:

1. Buka dokumen CDM yang sudah dibuat
2. Yakinkan bahwa diagram CDM yang telah dibuat sudah final tidak terdapat perubahan lagi.

3. Cek diagram model dengan menekan tombol F4. Pastikan bahwa hasil pengecekan menampilkan error: 0 dan warning : 0
4. Untuk membuat diagram PDM klik menu tools → klik generate physical data model (ctrl+shift+P )

5. Dukumen PDM hasil dari generate CDM dapat diedit kembali dan dapat ditambahkan obyek-obyek yang tersedia dalam jendela *pallete.Atribute atribute tambahan pada relasi dalam diagram CDM harus ditambahkan secara manual pada diagram PDM*
6. Sebelum mengkonversi dokumen CDM ke PDM, yakinkan bahwa semua obyek-obyek telah ditambahkan dan hasil check model tidak terdapat *error dan warning*.

Gambar dibawah ini merupakan salah satu contoh dokumen PDM dari konversi dokumen CDM.

## c. Rangkuman

PDM adalah merupakan suatu model implementasi phisik dari database. Dengan PDM, pengguna dapat mempertimbangkan secara detil tentang implementasi phisik nyata PowerDesigner adalah software tools dengan pendekatan model driven berbasis grafis yang digunakan untuk untuk menyelaraskan bisnis dan Teknologi informasi. Tools ini merupakan *enterprise modelling* yang membantu mengimplementasikan Enterprise Architecture dan membawa lingkungan manajemen meta-data yang kuat untuk siklus hidup pengembangan aplikasi. Berbagai jenis ragam model yang dapat dibuat menggunakan power designer adalah 1) Model Persyaratan (Requirements Model / RQM). 2) Proses Model Bisnis (Bussiness process Model / BPM). 3) Model Data Konseptual (*Conceptual Data Model* / CDM). 4) Physical Data Model ( PDM ). 5) Model data logis (Logical Data Model / LDM ). 6) Model Likuiditas Informasi (ILM). 7) Object-Oriented Model (OOM). 8) XML Model (XSM). 9) Model bebas (Free model/FEM).

## d. Tugas : Membuat physical data model (PDM)

Dalam kegiatan ini peserta didik akan melakukan praktikum secara berkelompok, satu kelompok terdiri dari dua sampai tiga orang. Peserta didik akan merancang

pemodelan data level fisik (physical data model). Bacalah seluruh langkah pengamatan dibawah ini kemudian lakukan dengan cermat dan teliti dengan perangkat yang telah disediakan.

1. Bentuk kelompok diskusi setiap kelompok terdiri dari tiga orang.
2. Instal aplikasi program sybase powerdesigner 12.5 atau yang terbaru. Amati setiap tampilan atau perubahan dilayar monitor dan catat hasilnya.
3. Buatlah diagram PDM menggunakan fungsi *new model*. Tambahkan semua obyek (sistem database perusahaan) meliputi: tabel, atribute sesuai dengan algoritma mapping ER to table pada kegiatan belajar 5. Catat hasilnya.
4. Berdasarkan uraian dari materi pembelajaran buatlah diagram PDM dengan mengkonversi dokumen CDM *(generate PDM*) untuk sistem database perusahaan. Tambahkan beberapa obyek atribute yang dibutuhkan. Atur semua propertis sesuai dengan kebutuhan sistem basis data. Catat hasilnya.
5. Kumpulkan data-data setiap langkah dan analisis data tersebut menggunakan analisis diskriptif. Bandingkan hasil praktek untuk langkah 3 dan 4.
6. Komunikasikan hasilnya dalam kelompok dan buatlah kesimpulan.
7. Presentasikan hasil diskusi bersama-sama dengan kelompok lainnya dan guru pembimbing.

**e. Test Formatif.**

Dalam test ini setiap peserta didik membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

1) Sebutkan dan jelaskan secara singkat fungsi atau kegunaan dari PDM ?
2) Sebutkan obyek-obyek atau komponen dalam Physical data model beserta fungsi dan kegunaannya?

**f. Lembar Jawaban Test Formatif (LJ).**

**LJ- 01 : Ragam jenis model diagram dalam tools power designer.**

**LJ- 02 : berbagai ragam jenis, relasi, cardinality dan contoh relasi antar entitas**

### g. Lembar Kerja Peserta Didik.

## Kegiatan belajar 13: Pengenalan SQL

**a. Tujuan Pembelajaran.**

Setelah mengikuti kegiatan belajar10 ini diharapkan peserta didik dapat :

- ✓ Memahami konsep bahasa Standar Query Language
- ✓ Mengoperasikan bahasa SQL data definition language (DDL)

**b. Uraian materi.**

**1. Definsi SQL**

SQL (Structured Query Language) adalah sebuah bahasa yang digunakan untuk mengakses data dalam software DBMS. Bahasa ini merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua server basis data mendukung bahasa ini untuk melakukan pengelolaan datanya. Instruksi – instruksi atau pernyataan SQL dapat dikelompokkan menjadi 5 kelompok DDL, DML, DCL,pengendali transaksi dan pengendali programatik.

- **DDL ( Data Definition Language )**

  DDL merupakan kelompok perintah yang berfungsi untuk mendefinisikan atribut-atributdatabase, table, atribut (kolom), batasan-batasan terhadap suatu atribut serta hubunganantar table. Yang termasuk dalam kelompok DDL ini adalah :

  *CREATE* untuk menciptakan table ataupun indeks

  *ALTER* untuk mengubah struktur table

  *DROP* untuk menghapus table ataupun indeks

- **DML ( Data Manipulation Language )**

  Adalah kelompok perintah yang berfungsi untuk memanipulasi data, misalnya untuk pengambilan, penyisipan pengubahan dan penghapusan data. Yang termasuk DML adalah instruksi:

  *SELECT* untuk memilih data (*retrieving* data)

  *INSERT* untuk menambah data

  *DELETE* untuk menghapus data

  *UPDATE* untuk mengubah data

- **DCL ( Data Control Language )**

  Berisi perintah-perintah untuk mngendalikan pengaksesan data. Yang termasuk DCL diantaranya adalah :

  | | |
  |---|---|
  | *GRANT* | untuk memberikan kendali pada pengaksesan data. |
  | *REVOKE* | untuk mencabut kemampuan pengaksesan data |
  | *LOCK TABLE* | untuk mengunci tabel |

- ***Transaction Control Language* (TCL) atau Pengendali transaksi**

  TCL adalah perintah-perintah yang berfungsi untuk mengendalikan pengeksekusian transaksi. Yang termasuk kelompok TCL ini adalah :

  | | |
  |---|---|
  | *COMMIT* | untuk menyetujui rangkaian perintah yang berhubungan erat yang telahberhasil dilakukan |
  | *ROLLBACK* | untuk membatalkan transaksi yang dilakukan karena adanya kesalahanatau kegagalan pada salah satu rangkaian perintah. |

**2. Mendesain Tabel dengan Query**

Suatu file database (*.mdb, *.accdb) terdiri dari satu atau lebih table, index dan komponen lainnyanya. Sedangkan dalam satu tabel bis aterdiri dari satu atau lebih record data masing-masing berisi informasi yang sejenis. Membuat database berarti membuat file pada disk dimana kita tidak dapat berbuat apa-apa dengan file tersebut sampai tabel-tabel selesai dibuat dan ditambahkan pada file database. Dalam mendesain tabel dengan Query pastikan jendela SQL Query aktif. Format perintah query sebagai berikut:

**CREATE TABLE NamaTabel (Field1 Type**
**[(Size)][NOT NULL][Index1][,Field2 Type**
**[(Size)][NOT NULL][Index2][,…]]**
**[,CONSTRAINT Multifieldindex[,…]])**

Keterangan :

| **Komponen** | Keterangan |
|---|---|
| Tabel | Nama dari tabel yang akan dibuat. |
| Field1, Field2 | Nama dari masing-masing field yang akan digunakan pada tabel yang baru dibuat. Anda harus membuat minimal satu field. |
| Type | Tipe data dari field yang digunakan pada tabel baru. |

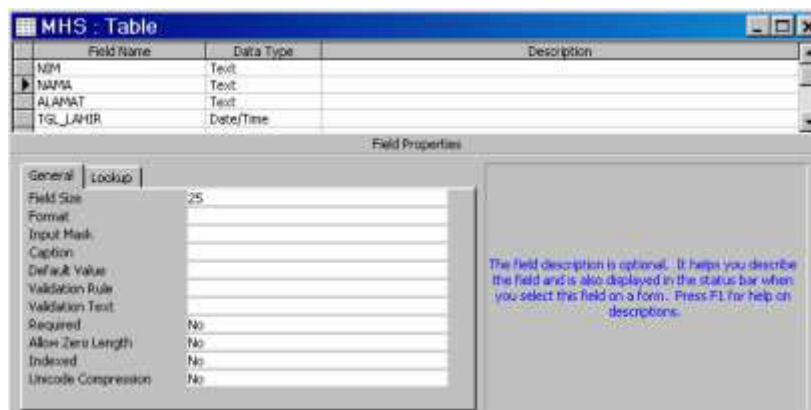
| Size | Ukuran dari field dalam karakter. Digunakan hanya untuk tipe data Text. |
|---|---|
| Index1, Index2 | Anak kalimat Contraint yang mendefinisikan sebuah index Single field. |
| Multifieldindex | Anak kalimat Contraint yang mendefinisikan sebuah index Multiple field. |

Contoh dari perintah query yang digunakan untuk membuat tabel MHS sebagai berikut :

```
CREATE TABLE MHS (
NIM TEXT(7),
NAMA TEXT(25),
ALAMAT TEXT(40),
TGL_LAHIR   DATE )
```

Beberapa elemen umum yang harus ditentukan dalam membuat sebuah tabel:

- Nama dari tabel harus Unique untuk setiap file database, tidak diperkenankan dalam satu folder terdapat lebih dari satu nama file database yang sama.
- Nama dari field (kolom) harus bersifat Unique untuk setiap tabel (tidak boleh sama).
- Tipe data dan ukuran masing-masing field (kolom) harus disesuaikan dengan kondisi data yang akan disimpan.
- Pemakaian Constraint yang diikutkan dalam pembentukan suatu tabel, terdiri dari Null, Not Null, Primary Key, Unique dan Foreign Key atau gabungan beberapa Constraint yang ada.

Hasil dari statement query stuktur tabel MHS diatas adalah sebagai berikut :

### 3. Modifikasi Tabel Dengan Query

Keberadaan suatu tabel yang telah dibuat tidak selamanya akan seperti yang tilah kita buat, terkadang pemakai ingin mengadakan perubahan atau modifikasi tabel tersebut baik yang berkaitan dengan struktur maupun hal lain seperti : manambah, mengubah dan mengahapus batasan dan sebagainya.

Format penulisan untuk memodifikasi suatu tabel:

**ALTER TABLE Nama Tabel {ADD(COLUMN Field1 Type [(Size)] [NOT NULL] [CONSTRAINT Index][CONSTRAINT Multifieldindex} | DROP {COLUMN Field|CONSTRAINTNamaConstraint}}**

Keterangan :

| Komponen | Keterangan |
|---|---|
| FieldType | Tipe data dari field yang digunakan. |
| Multifieldindex | Definisi multiple field index yang akan ditambahkan ke dalam tabel. |
| NamaConstraint | Nama constraint yang akan dihapus. |

Beberapa contoh query yang berhubungan dengan modifikasi data tabel :

- Perintah untuk menambah field (kolom) baru yaitu JENIS_KEL dengan tipe data Text sebesar 1 karakter.

  **Alter Table MHS Add JENIS_KEL Text(1)**
- Perintah untuk merubah ukuran dan tipe data dari suatu field NIM dengan tipe data Number.

  **Alter Table MHS Alter Column NIM Number**
- Perintah untuk menghapus field (kolom) JENIS_KEL.

  **Alter Table MHS Drop Column JENIS_KEL**

Add atau Drop tidak dapat diterapkanpada field yang jumlahnya lebih dari satu pada waktu yang bersamaan.

### 4. Menghapus Tabel Dengan Query

Sebuah table yang tidak digunakan lagi dapat dihapus dari suatu file database dengan perintah query. Seperti halnya menghapus suatu field dalam suatu tabel maka Format penulisan untk menghapus tabel adalah :

**DROP {TABLE NamaTabel | INDEX NamaIndex ON Indextable}**

Keterangan :

| Komponen | Keterangan |
|---|---|
| Nama table | Nama tabel yang akan dihapus |
| NamaIndex | Nama index yang akan dihapus. |
| Indextable | Nama tabel yang indexnya akan dihapus. |

Beberapa ketentuan yang harus diperhatikan dalam melakukan penghapusan tabel:

- Sebelum tabel dihapus, Anda harus menutup terlebih dahulu tabel tersebut.
- Suatu tabel yang telah dihapus tidak dapat dikembalikan seperti semula.
- Data record yang ada pada tabel yang dihapus juga akan terhapus.
- Tidak ada pesan/persetujuan terlebih dahulu selama proses penghapusan tabel dilaksanakan.

**5. Pemakaian Constraint**

Constraint merupakan istilah yang digunakan untuk menerapkan Integritas Data (*Data Integrity)* pada suatu database. Integritas Data merupakan istilah yang digunakan untuk menggambarkan kebenaran data di dalam suatu file database. Sedapat mungkin integritas data tetap dijaga melalui beberapa kegiatan, antara lain:

- Validasi field secara individual
- Verifikasi satu field melalui field lainnya
- Validasi data dari tabel satu ke tabel lainnya
- Verifikasi bahwa transaksi berjalan sukses dari awal hingga akhir

Terdapat tiga jenis Integritas Data yang dapat diterapkan pada file database Microsoft Access 2002, antara lain:

- Integritas Entitas, untuk menerapkan integritas pada tingkat Entity (Tabel), agar setiap Instances (Record\Baris) pada suatu Entity bersifat Unique yang disebut sebagai Primary Key sehingga dapat dibedakan antara yang satu dengan lainnya. Hubungan antara primary key dan foreign key menyatakan apakah sebuah baris tabel dapat diubah atau dihapus.

- Integritas Domain, untuk menerapkan integritas pada tingkat Kolom (field) agar nilai yang diisikan pada Kolom (field) tersebut berada di antara nilai-nilai yang diinginkan.
- Integritas Referential untuk menerapkan integritaspada tingkat Referensi dengan menggunakan Primary Key untuk memastikan sinkronisasi antara "Parent" dan "Child Tables" (antara Primary Key dan Foreign Key). Sebuah Foreign Key tidak dapat dimasukkan ke dalam sebuah tabel, bila Primary Key tidak ada atau belum ada.

## 6. Constraint Primary Key

Primary Key disebut sebagai "Constraint" dengan tujuan untuk menjaga integritas data, yaitu bahwa sebuah Primary Key tidak boleh mempunyai "Duplikat" dan secara otomatis tidak "Null". Bentuk penulisan Constraint Key secara umum:

**CONSTRAINT NamaConstraint{PRIMARY KEY (Primary1 [, Primary2 [...] ] ) }**

Keterangan :

| Komponen | Keterangan |
|---|---|
| NamaConstraint | Nama dari constraint |
| Primary1, Primary2 | Nama dari masing-masing field yang digunakan sebagai key (kunci) |

Berikut format penulisan constraint Primary Key pada tabel MHS dengan Primary Key NIM dimana penulisannya diletakkan setelah field NIM dan tipe datanya, yaitu:

**Create Table MHS ( NIM TEXT(7) Constraint PK_NIM Primary Key, NAMA TEXT(25), ALAMAT TEXT(40), TGL_LAHIR DATETIME**)

Penulisan nama Primary Key dengan PK_NIM mempunyai makna PK adalah Primary Key dan NIM adalah nama field kunci.

MHS : Table

| Field Name | Data Type |
|---|---|
| NIM | Text |
| NAMA | Text |
| ALAMAT | Text |
| TGL_LAHIR | Date/Time |

Altrernatif lain penulisan Constraint dengan Primary Key NIM dimana constraintnya diletakkan pada bagian terakhir:

**Create Table MHS (**
**NIM TEXT(7),**
**NAMA TEXT(25),**
**ALAMAT TEXT(40),**
**TGL_LAHIR DATETIME,**
**Constraint PK_NAMA Primary Key (NIM) )**

Penulisan Primary Key dengan banyak field:

**Create Table MHS ( NIM TEXT(7),**
**NAMA TEXT(25),**
**ALAMAT TEXT(40),**
**TGL_LAHIR DATETIME,**
**Constraint PK_MHS Primary Key (NIM, NAMA) )**

Sedang bila Anda ingin menambahkan Primary Key pada tabel yang sudah terbentuk tetapi belum mempunyai Primary Key, format penulisannya seperti berikut:

**Alter Table MHS Add Constraint PK_NIM2 Primary Key (NIM)**

## 7. Constraint Unique

Nilai Unique digunakan untuk menjamin bahwa nilai dalam satu field adalah "Tunggal" (tidak mempunyai duplikat). Beberapa perbedaaan antara Constraint Primary Key dan Constraint Unique:

| Constraint Primary Key | Constraint Unique |
|---|---|
| Hanya diperbolehkan menggunakan satu Constraint Primary Key dalam satu tabel. | Boleh menggunakan lebih dari satu Constraint Unique dalam satu tabel. |
| Field (kolom) yang berpatisipasi, tidak boleh mengandung nilai NULL. | Field (kolom) yang berpatisipasi, boleh mengandung nilai NULL. |

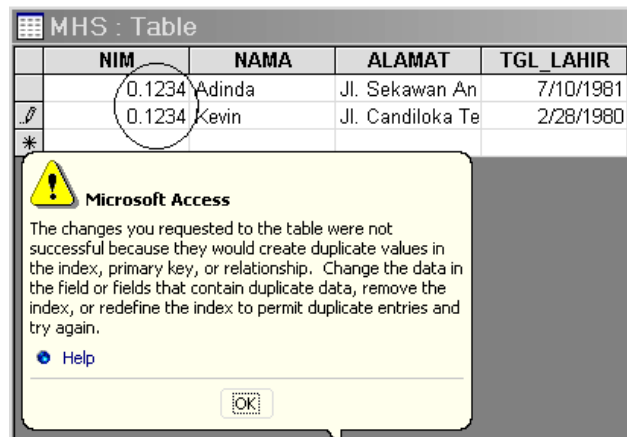
Format penulisan Constraint Unique secara umum:

**CONSTRAINT NamaConstraint {UNIQUE (Unique1 [, Unique2 [, …] ] ) }**

Keterangan :

| Komponen | Keterangan |
|---|---|
| Unique1, Unique2 | Nama dari masing-masing field yang dibuat Unique. |

Contoh format penulisan constraint Unique dengan Unique NIM:

**Create Table MHS (**
**NIM TEXT(7)Constraint UN_NIM Unique,**
**NAMA TEXT(25),**
**ALAMAT TEXT(40),**
**TGL_LAHIR DATETIME )**

Alternatif lain penulisan Constraint Unique untuk field NAMA:

**Create Table MHS (NIM TEXT(7), NAMA TEXT(25), ALAMAT TEXT(40), TGL_LAHIR DATETIME, Constraint UN_NAMA Unique (NAMA)** )

Untuk penulisan Constraint Unique dengan multifield:

**Create Table MHS (NIM TEXT (7), NAMA TEXT(25), ALAMAT TEXT(40), TGL_LAHIR DATETIME, Constraint UN_MHS Unique (NIM, NAMA) )**

Bila ingin menambahkan Constraint Unique pada tabel yang sudah terbentuk tetapi belum mempunyai file Unique:

**Alter Table MHS Add Constraint UN_NIM Unique (NIM)**

### 8. Constraint Not Null

Bentuk penulisan umum Constraint Not Null:

**NamaField NOT NULL**

Contoh format penulisan Constraint Not Null dengan field NIM yang dipasang Not Null:

**Create Table MHS (NIM TEXT(7) Not Null, NAMA TEXT(25), ALAMAT TEXT(40), TGL_LAHIR DATETIME)**

Default penetapan seluruh field yang dibuat pada suatu tabel adalah Null (kosong), artinya Anda diperbolehkan untuk tidak mengisi suatu field, meskipun field tersebut bersifat Unique. Tetapi bila field yang bersangkutan berfungsi sebagai Primary Key, maka field tersebut tidak boleh kosong.

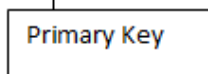
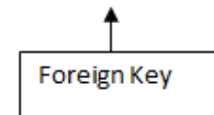
### 9. Constraint Foreign Key

Foreign Key (FK) adalah field pada sebuah tabel yang menunjukan bahwa field tersebut adalah Primary Key pada tabel yang lain. Untuk memperjelas makna perhatikan skema berikut ini.

Tabel A

| Field 1 | Field 2 | Field 3 | Field 4 |
|---|---|---|---|
| 001 | aaa | bbb | 234 |
| 002 | mmm | nnn | 173 |

Tabel B

| Filed 1 | Field 2 |
|---|---|
| 173 | XXX |
| 234 | ZZZ |



Pada skenario di atas, record (173, XXX) pada tabel B tidak dapat dihapus karena referensi dari 173, fild 4 di Tabel A masih ada. Untuk dapat menghapus record tersebut, maka hapus dulu semua record di filed 4 pada tabel A yang mempunyai nilai 173. Foreign Key harus menunjuk ke Primary Key atau Unique pada tabel lain. Format penulisan Constraint Foreign Key:

**CONSTRAINT NamaConstraint {FOREIGN KEY (Ref1[ , Ref2[ , ... ] ] ) REFERENCES TabelForeign [ (Field1Foreign [ , Field2Foreign [ , ... ] ] ) ] }**

Keterangan :

| Komponen | Keterangan |
|---|---|
| Ref1, Ref2 | Nama dari masing-masing field yang digunakan sebagai key (kunci). |
| TabelForeign | Nama tabel yang digunakan sebagai rujukan. |
| Field1Foreign, Field2Foreign | Nama field dari tabel rujukan yang digunakan sebagai kunci penghubung dengan tabel yang sedang dibuat. |

Contoh dari format penulisan constraint Foreign Key pada tabel JUAL dengan Primary Kd_trans dengan tabel rujukan JENIS dengan field kunci Kode. Pertama-tama buat tabel yang digunakan sebagai rujukan yaitu tabel JENIS dengan field Kode dan Nama, seperti pada contoh berikut ini:

**Create Tabel JENIS (Kode Text(5) Constraint PK_Kode Primary Key, Nama Text(50) )**

Selanjutnya bisa dibuat tabel baru dengan nama JUAL dimana tabel ini dibnetuk dengan merujuk tabel yang sudah ada yaitu tabel JENIS. Untuk merujuk digunakan perintah "References" seperti contoh berikut ini:

**Create Table JUAL (No_Trans Text(7), Tgl_trans Date, Kd_Trans Text(5), Jml_Trans Long, Constraint F_Jual Foreign Key (Kd_Trans) References JENIS (Kode) )**

JENIS : Table

| KODE | NAMA |
|---|---|
| EL123 | Televisi 20" |
| EL987 | Radio-Tape |
| PK123 | T-Shirt |
| PK987 | Sepatu Bola |

Record: 1 of 4

JUAL : Table

| NO_TRANS | TGL_TRANS | KD_TRANS | JML_TRANS |
|---|---|---|---|
| 1000001 | 8/15/2001 | EL123 | 2 |
| 1000002 | 8/17/2001 | PK123 | 1 |

Record: 1 of 2

Jika kode barang yang diinputkan pada tabel JUAL tidak ada di tabel JENIS maka proses input data akan ditolak, hal ini dikarenakan nilai dari field "Kd_Trans" sudah direferensikan ke field "Kode" yang ada di tabel JENIS.

Gambar dibawah ini menjelaskan 7 Pesan yang Menyatakan Pembatalan Proses Input Data pada Tabel JUAL yang Dilakukan dari Datasheet dan Pesan yang Menyatakan Pembatalan Proses Input Data pada Tabel JUAL yang Dilakukan dengan Query

Kelebihan pemakaian Constraint Foreign Key adalah pada tabel referensi terdapat tanda plus (+) di paling kiri dari masing-masing recordnya, tanda ini menyatakan bahwa record-record tersebut telah direferensikan ke tabel lain. Lakukan klik pada tanda plus maka akan ditampilkan record-record dari hasil referensi. Gambar dibawah ini menjelaskan Data-data yang Direferensikan ke Kode Barang EL123

## c. Rangkuman

SQL (Structured Query Language) adalah sebuah bahasa yang digunakan untuk mengakses data dalam software DBMS. Bahasa ini merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua server basis data mendukung bahasa ini untuk melakukan pengelolaan datanya. Instruksi – instruksi atau pernyataan SQL dapat dikelompokkan menjadi 5 kelompok DDL, DML, DCL,pengendali transaksi dan pengendali programatik.

*Data Definition Language* (DDL merupakan kelompok perintah yang berfungsi untuk mendefinisikan atribut-atribut database, table, atribut (kolom), batasan-batasan terhadap suatu atribut serta hubungan antar table. Yang termasuk dalam kelompok DDL ini adalah : 1) *CREATE* untuk menciptakan table ataupun indeks 2). *ALTER* untuk mengubah struktur table 3) *DROP* untuk menghapus table ataupun indeks.

Kode program dalam bahasa SQL untuk membuat tabel, view dan relasi antar tabel dapat dibuat secara oromatis (generate script SQL) dari diagram PDM menggunakan alat bantu software power designer. Hasil dari generate script SQL adalah sebuah file dengan ekstensi (*.sql). Selanjutnya script SQL tersebut dapat di-*execute* pada DBMS sesuai dengan target database yang telah ditentukan sebelum proses *generate script* SQL.

**d. Tugas : Membuat Script SQL**

Dalam kegiatan ini peserta didik akan melakukan praktikum secara berkelompok, satu kelompok terdiri dari dua sampai tiga orang. Peserta didik akan menuliskan script SQL untuk membangun database relasional sesuai diagram relasi entitas (ERD) pada kegiatan belajar 12. Bacalah seluruh langkah pengamatan dibawah ini kemudian lakukan dengan cermat dan teliti dengan perangkat yang telah disediakan.

1. Bentuk kelompok setiap kelompok terdiri dari dua-tiga orang
2. Berdasarkan hasil kegiatan belajar 12 (dokumen *physical data model*), dengan menggunakan alat tulis dan kertas pada lembar kerja tulislah script SQL untuk membuat tabel-tabel tersebut. Tulis dan catat hasilnya.
3. Kumpulkan data-data setiap langkah dan analisis data tersebut menggunakan analisis diskriptif.
4. Komunikasikan hasilnya dalam kelompok, buatlah kesimpulan dan laporan sementara.
5. Komunikasikan dan verifikasi hasil kepada guru pembimbing.
6. Lampirkan hasil verifikasi dan pengesahan laporan sementara ke dalam laporan lengkap.

**e. Test Formatif.**

Dalam test ini setiap peserta didik membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

1. Sebutkan dan jelaskan secara singkat lima ragam jenis (kelompok) instrukti-instruksi dalam SQL ?
2. Sebutkan dan berikan contoh instruksi –instruksi SQL yang termasuk dalam kelompok DDL (data definition language)ty relasi tersebut.?

**f. Lembar Jawaban Test Formatif (LJ).**

**LJ- 01 : Ragam jenis (kelompok) instruksi-instruksi dalam SQL.**

........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................

**LJ- 02 : Ragam jenis Instruksi DDL SQL dan contohnya**

**g. Lembar Kerja Peserta Didik.**

**Kegiatan belajar 14: Mengoperasikan SQL dalam DBMS**

**a. Tujuan Pembelajaran.**

Setelah mengikuti kegiatan belajar10 ini diharapkan peserta didik dapat:

- ✓ Menuliskan perintah SQL dalam jendela SQL Veiw MS Access
- ✓ Menjalankan Perintah SQL dalam jendela SQL Veiw MS Access
- ✓ Memperbaiki kesalahan perintah SQL

**b. Uraian materi.**

1. **Sekilas Microsoft Access 2010**

Microsoft Access merupakan salah satu program pengolah data base yang digunakan untuk mengolah berbagai jenis data dengan pengoperasian yang mudah. Kemudahan yang ada dalam microsoft access antara lain ialah proses penyortiran, pengaturan, pembuatan Tabel data serta Ipembuatan laporan data kegiatan sehari-hari misalnya untuk menampung daftar pelanggan, pendataan data karyawan. Microsoft Access 2010 memperkenalkan berbagai fitur yang tidak didukung dalam MS Access 2007. MS 2010 menawarkan beberapa perbaikan yang membuat aplikasi lebih bermanfaat, termasuk perangkat tambahan kehandalan. Gambar dibawah ini menjelaskan berbagai ragam icon toolbars micrpspft access 2010.

2. **Ragam tipe data dalam MS Access**

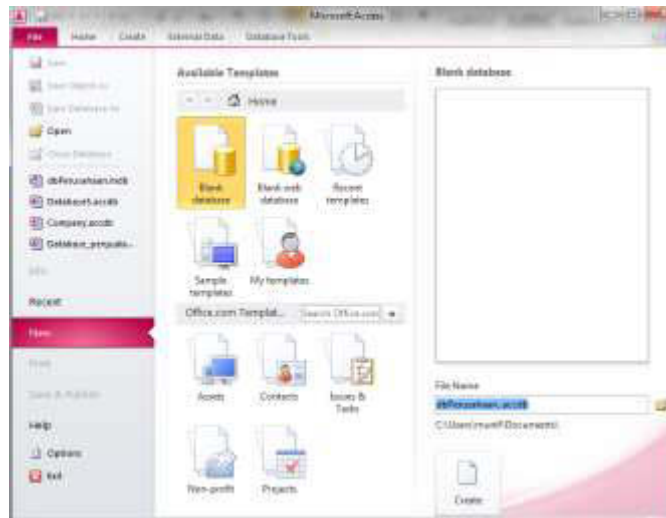
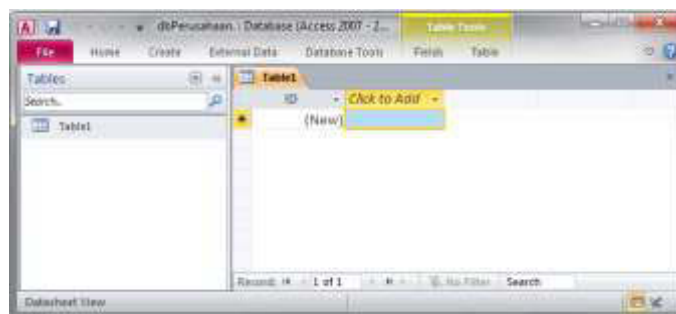
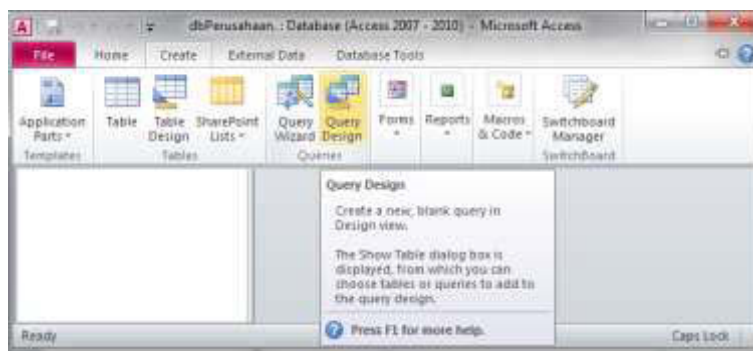
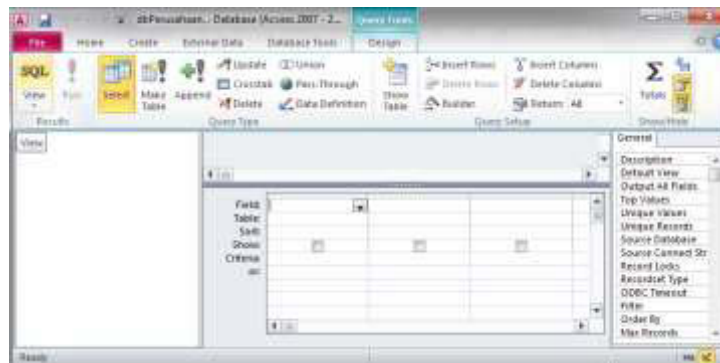
Berbagai ragam jenis tipe data beserta ukurannya dalam microsoft access 2010 antara lain adalah sebagai berikut:

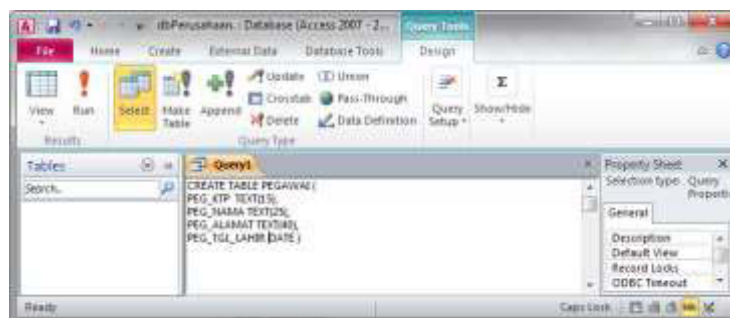
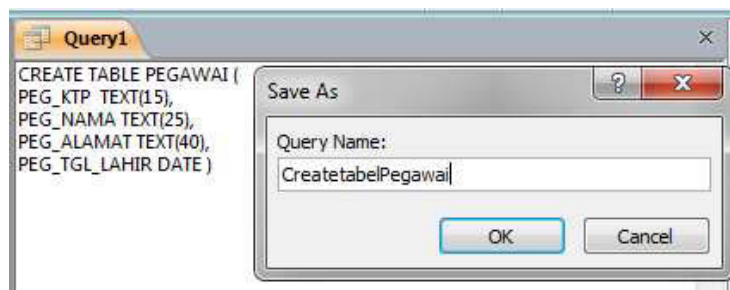
1. Text. Pada tipe data ini jenis data yang disimpan adalah karakter. Panjang maksimal *type field* adalah 255 karakter yang merupakan *type default*.
2. Memo. Pada tipe data ini, jenis data yang disimpan adalah karakter. Panjang maksimal *type field* adalah 65.535 karakter.
3. Number Merupakan tipe data yang digunakan untuk menampung *type data* angka.
4. Date/time. Jenis data yang disimpan adalah data tanggal dan waktu dengan besar memory 8 *byte*.
5. Currency. Merupakan tipe data yang digunakan untuk menyimpan angka dalam format mata uang. Besarnya memori penyimpanan adalah 4 *byte*.
6. Auto Number. Tipe data ini digunakan untuk memberikan penomoran secara otomatis (penambahan angka otomatis)
7. Yes/No. Tipe data ini berisikan data *Yes* atau *No*, Benar atau Salah, Ya atau Tidak.
8. OLE Object. Tipe data ini dapat memuat gambar, grafis, video dan suara dengan ukuran maksimal 1 GB (batas atas *Harddisk*)
9. Hyperlink. Tipe data yang berisikan alamat *hyperlink URL* dengan panjang maksimal 64.000 karakter. Lookup Wizard. Tipe data yang digunakan untuk menampilkan data dari tabel lain. Besar memori penyimpanan umumnya 4 *byte*.

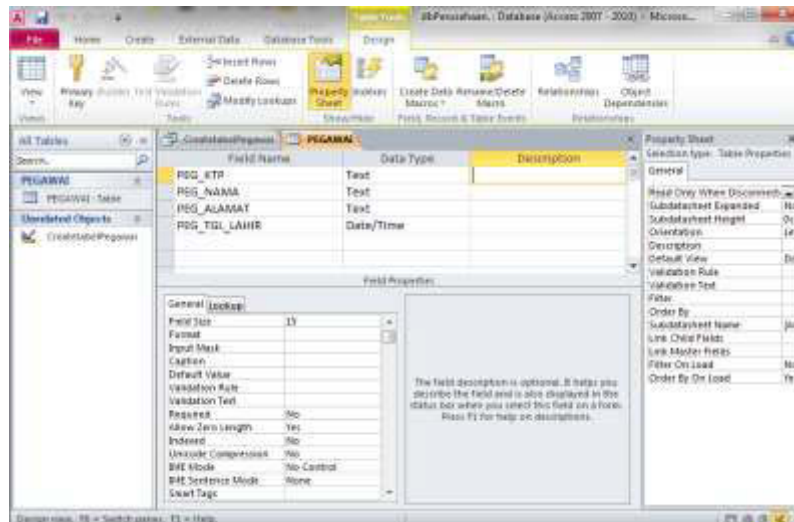
3. **Membuat database**

Langkah-langkah yang dilakukan untuk membuat basis data dalam microsoft access 2010 adalah sebagai beikut:

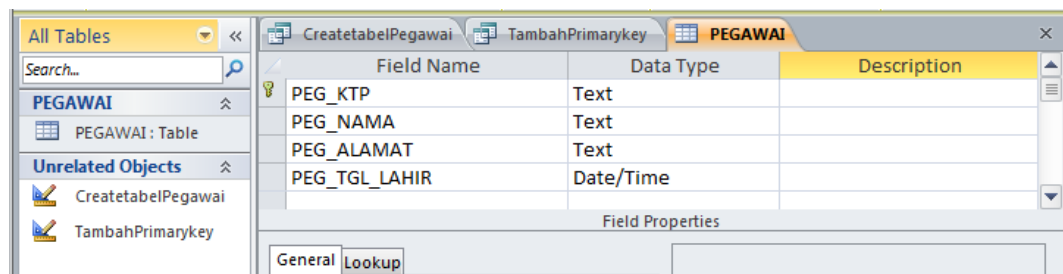
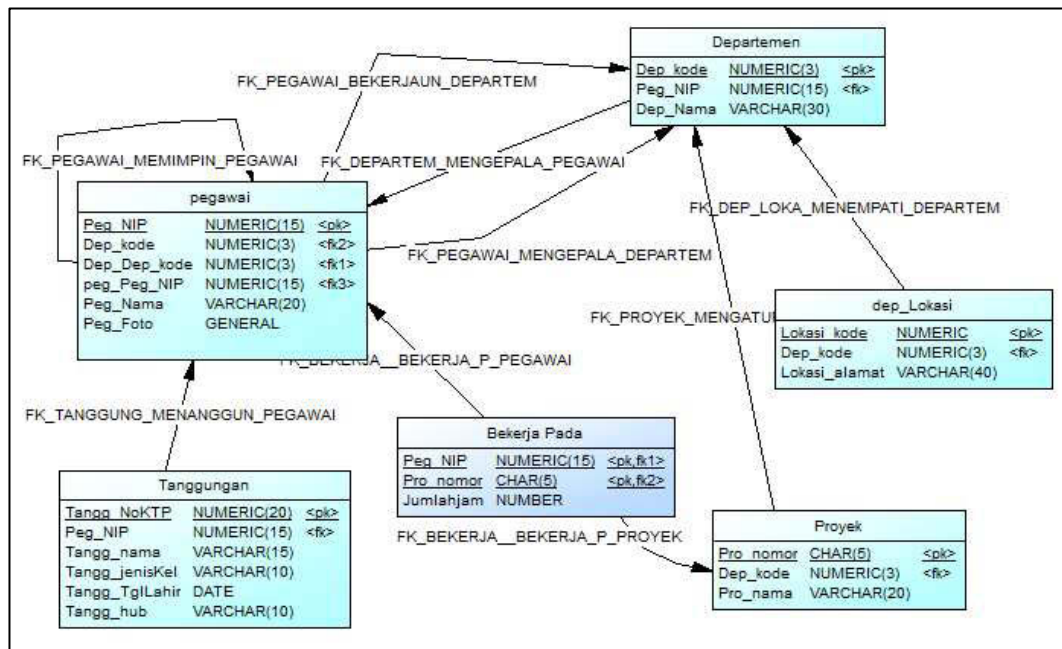
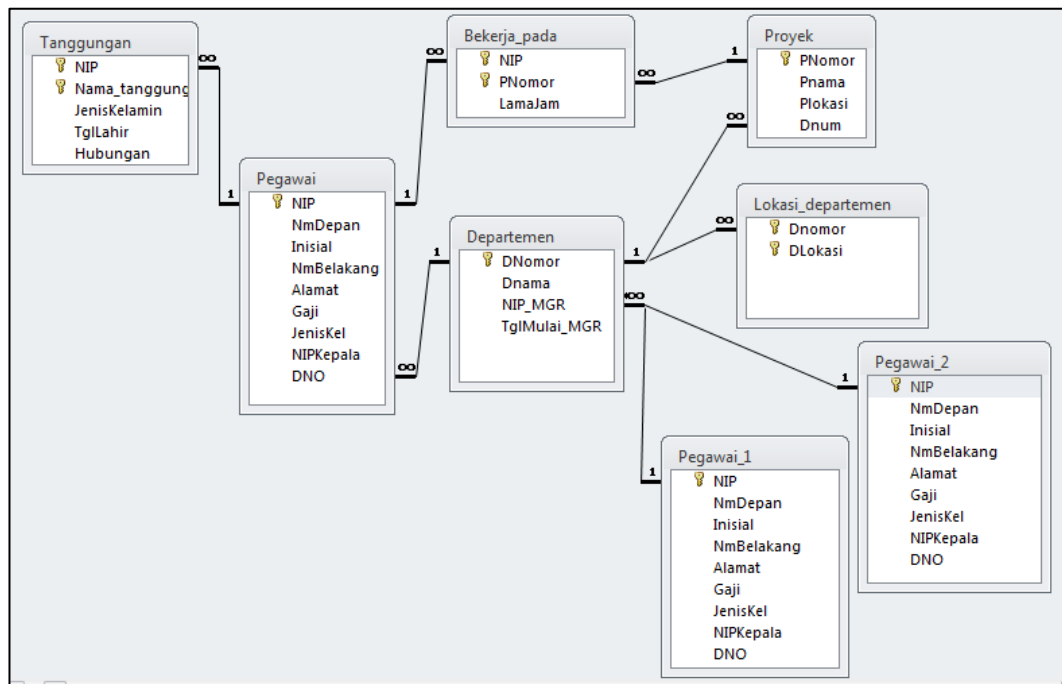
**a.** Buka program aplikasi microsoft access 2010.

b. Klik new tulis nama file database misal : dataBasepersuahaan

c. Pilih folder untuk menyimpan kemudian klik tombol create

d. Secara otomatis terbentuk satu tabel, hapus tabel tersebut dengan klik kanan nama tabel kemudian klik delete.

e. Pilih menu create klik query design maka akan muncul jendela query dan show tabel→ klik close untuk menutup jendela show tabel

f. Pada jendela query pilih menu Design dan klik icon SQL view pada toolbars

g. Tuliskan script SQL untuk membuat tabel pegawai

h. Klik kanan query1 kemudian klik save tulis nama file kemudian klik oke

i. Klik icon run untuk menjalankan perintah SQL.

j. Klik kanan pada tabel pegawai kemudian klik design view untuk melihat jendela desaian tabel

k. Dari gambar terlihat bahwa tabel diatas belum mempunyai primary key.

l. Untuk menambahkan primary key, buatlah query dengan nama TambahPrimaryKey, tuliskan script SQL dibawah ini kemudian jalankan perintah SQL tersebut

   Alter Table PEGAWAI Add Constraint PK_KTP Primary Key (PEG_KTP)

m. Ulangi langkah e untuk tabel lainnya berdasarkan pemodelan data fisik (PDM) dalam kegiatan belajar 12 seperti terlihat dalam gambar dibawah ini..

n. Jika seluruh tabel sudah terbuat. Tampilkan relasi tabel tersebut dengan menekan menu databasetools kemudian klik relationship. Gambar dibawah ini menjelaskan contoh relasi tabel yang telah dibuat. Terdapat beberapa perbedaan untuk Nama field pada tabel yang terbentuk terhadap rancangan yang dibuat.

**c. Rangkuman**

Microsoft Access merupakan salah satu program pengolah data base yang digunakan untuk mengolah berbagai jenis data. Kemudahan yang ada dalam microsoft access antara lain ialah proses penyortiran, pengaturan, pembuatan Tabel data serta Ipembuatan laporan data kegiatan sehari-hari misalnya untuk menampung daftar pelanggan, pendataan data karyawan.

**d. Tugas : Mengoperasikan perintah SQL**

Dalam kegiatan ini peserta didik akan melakukan praktikum secara berkelompok, satu kelompok terdiri dari dua sampai tiga orang. Peserta menjalankan script SQL yang telah dibuat dalam kegiatan belajar 13 pada DBMS microsoft access.. Bacalah seluruh langkah dibawah ini kemudian lakukan dengan cermat dan teliti dengan perangkat yang telah disediakan.

1. Bentuk kelompok diskusi setiap kelompok terdiri dari dua-tiga orang.
2. Jalankan dan buka aplikasi microsoft acsess. Buatlah database dengan nama dbPerusahaan.accdb.
3. Buatlah query tuliskan script SQL untuk membuat seluruh tabel beserta relasi tabel untuk data base perusahaan sesuai dengan relasi tabel physical data model (PDM) dalam kegiatan belajar 12. Beri nama dan simpan setiap query yang dibuat.
4. Jalankan script SQL setiap query yang dibuat. Catat hasilnya dengan meng-capture setiap tabel dalam fungsi design view
5. Kumpulkan data-data setiap langkah dan analisis data tersebut menggunakan analisis diskriptif.
6. Komunikasikan hasilnya dalam kelompok, buatlah kesimpulan dan laporan sementara.
7. Komunikasikan dan verifikasi hasil kepada guru pembimbing.
8. Lampirkan hasil verifikasi dan pengesahan laporan sementara ke dalam laporan lengkap.

**e. Test Formatif.**

Dalam test ini setiap peserta didik membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

1. Sebutkan dan jelaskan berbagai ragam tipe data dalam microsoft access 2010 ?
2. Sebutkan langkah-langkah membuat query yang digunakan untuk menuliskan scrip SQL dan menjalankan script SQL pada microsoft access.?

**f. Lembar Jawaban Test Formatif (LJ).**

**LJ- 01 : Ragam jenis Tipe data dalam microsoft access 2010 ?**

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

..................................................................................................................

**LJ- 02 : Langkah-langkah membuat query, menuliskan script SQL dan menjalankannya untuk membuat tabel**

7. **Lembar Kerja Peserta Didik.**

## Kegiatan belajar 15: Perintah SQL: Modifikasi Data

### a. Tujuan Pembelajaran.

Setelah mengikuti kegiatan belajar 15 ini diharapkan peserta didik dapat:

- ✓ Mengoperasikan perintah Insert Into untuk menambahkan data
- ✓ Mengoperasikan perintah Update untuk merubah data
- ✓ Menggabungkan perintah select dan insert into
- ✓ Mengoperasikan perintah delete untuk menghapus data

### b. Uraian materi.

Modifikasi data atau manipulasi data merupakan kegiatan yang bertujuan untuk mengubah data-data yang ada pada suatu tabel. Perubahan-perubahan data tersebut meliputi :manambah data, mengganti data maupun menghapus data. Tujuan perubahan tersebut adalah untuk mendapatkan data yang sesuai dengan lebutuhan pemakai (user)

#### 1) Menambah Data Baru

Menambah data baru kedalam suatu tabel dapat dilakukan dengan membuat tabel terlebih dahulu dimana data-data tersebut ditempatkan.

#### a) Pemakaian Perintah Insert Into

Salah satu perintah yang digunakan untuk menambahkan data baru pada suatu table adalah INSERT INTO dengan format penulisan sebagai berikut :

- Menambah satu record (Single-record Append Query)

  **INSERT INTO target [ (field1 [ , filed2 [ , … ] ] ) ] VALUES (value1 [ , value2 [ ,…])** Atau
  **INSERT INTO target [ (field1 [ , field2 [ , … ] ] ) ] SELECT value1 [ , value2 [ , …];**

- Menambah banyak record (Multiple-record Append Query)

  **INSERT INTO target [IN externaldatabase] [ (field1 [ , field2 [ , … ] ] ) ] SELECT [source.] field1 [ , field2 [ , … ] FROM table expression**

**Keterangan :**

| **Komponen** | **Keterangan** |
|---|---|
| Target | Nama dari tabel yang akan ditambahkan recordnya. |
| Field1, field2 | Nama dari masing-masing field yang ke dalamnya expresi value1…value2 akan diisikan. |

| | |
|---|---|
| Value1, Value2 | Ekspresi yang akan dimasukkan ke dalam field1…field2. Setiap ekspresi harus memiliki pasangan dengan field dan akan ditempatkan berdasarkan urutan penulisannya yaitu field1 = value1, field2 = value2, dan seterusnya. Bila antara field dan value tidak berhubungan maka perintah Insert akan gagal. |
| Externaldatabase | Nama database eksternal lengkap dengan direktorinnya. |
| Source | Nama dari tabel atau query yang digunakan sebagai tempat menampung data. |
| Tableexpression | Nama dari tabel dimana datanya akan diambil. Datanya dapat terdiri dari satu tabel atau join beberapa tabel. |
| Fieldlist | Nama-nama field yang akan diambil dengan perintah Select. |

- **Menambah satu record (Single-record Append Query)**

  Yang perlu diperhatikan dalam menambah record baru adalah data yang ingin ditambahkan harus mempunyai tipe data yang sama dengan dimana data tersebut akan ditampung. Sebagai gambaran akan dibuat struktur tabel baru dengan nama MHS_3:

  **Create Table MHS_3 (NIM Text(7), Nama Text(30), Tgl_Lahir Date, Danem Integer)**

  Sekarang ketikkan perintah baru untuk memasukkan data:

  **Insert Into MHS_3 (NIM, NAMA, TGL_LAHIR, DANEM) Values (“0012345”, “Adinda”, “17/03/1981”, 48)**

  Atau

  **Insert Into MHS_3 Values (“0012345”, “Adinda”, “17/03/1981”, 48)**

  Cara lain memasukkan data:

  **Insert Into MHS_3 (NIM, NAMA, TGL_LAHIR, DANEM) Select “0012345”, “Adinda”, “17/03/1981”, 48**

  Bila data yang akan dimasukkan hanya ke beberapa field saja maka nama field yang bersangkutan harus disertakan, penulisannya:

  **Insert Into MHS_3 (NIM, NAMA) Values (“0012345”, “Adinda”)**

MHS_3 : Table

| NIM | Nama | Tgl_Lahir | Danem |
|---|---|---|---|
| 0012345 | Adinda | 3/17/1981 | 48 |
| | | | |

Record: 1 of 1

- **Menambah banyak record (Multiple-record Append Query)**

  Cara memasukkan data dari tabel satu ke tabel lain, misalnya tabel baru yang akan kita jadikan target adalah DAFTAR_MHS, maka kita harus membuat terlebih dahulu tabel tersebut:

  **Create Table DAFTAR_MHS (NIM_MHS Text(7), NAMA_MHS Text(30) )**

  Selanjutnya data-data yang ada di field NIM dan NAMA pada tabel MHS_3 akan diduplikat ke tabel DAFTAR_MHS, perintahnya:

  **Insert Into DAFTAR_MHS (NIM_MHS, NAMA_MHS) Select NIM, NAMA From MHS_3**

DAFTAR_MHS : Table

| NIM_MHS | NAMA_MHS |
|---|---|
| 0012345 | Adinda |
| | |

**b) Pemakaian Perintah Select**

Perintah select digunakan untuk membentuk tabel baru dengan cara mengcopy seluruh data dari tabel yang aktif. Yang perlu diperhatikan adalah bila nama tabel hasil sudah pernah dibuat maka seluruh isi tabel dan strukturnya akan diganti dengan isi tabel dan struktur yang baru, format penulisannya:

**SELECT field1 [ , field2 [ , … ] ] INTO newtable [IN externaldatabase]FDRom source**

Keterangan :

| **Komponen** | **Keterangan** |
|---|---|
| Newtable | Nama tabel baru sebagai tempat hasil dari proses copy. |

Aktifkan tabel MHS_3 dan buatlah perintah query baru yang bertujuan untuk menduplikat seluruh data dari tabel MHS_3 ke tabel baru yang bernama MHS_BARU, format penulisannya:

**Select * Into MHS_BARU From MHS_3**

Bila hanya ingin menduplikat data untuk beberapa field, format penulisannya:

**Select NIM, NAMA Into MHS_BARU From MHS_3**

MHS_BARU : Table

| | NIM | Nama | Tgl_Lahir | Danem |
|---|---|---|---|---|
| ▶ | 0012345 | Adinda | 3/17/1981 | 48 |
| * | | | | |

Record: 1 of 1

**2) Mengubah Data**

Pada kondisi tertentu kita ingin mengubah salah satu atau lebih field yang terdapat pada satu atua lebih record. Perintah yang digunakan adalah UPDATE dengan Format penulisan sebagai berikut :

**UPDATE tabel SET field1=value1 [ , field2=value2 [ , fieldN=valueN] ] WHERE criteria**

| **Komponen** | **Keterangan** |
|---|---|
| Kriteria | Criteria dari baris data yang akan diubah. |

Sebelumnya kita tambahkan data pada tabel MHS_3, dengan menggunakan perintah Insert Into;

| | | | |
|---|---|---|---|
| **0012345** | **Adinda** | **17/03/1981** | **48** |
| **0012348** | **Kevin** | **28/02/1980** | **46** |
| **0012350** | **Putra** | **08/01/1983** | **40** |

MHS_3 : Table

| | NIM | Nama | Tgl_Lahir | Danem |
|---|---|---|---|---|
| ▶ | 0012345 | Adinda | 3/17/1981 | 48 |
| | 0012348 | Kevin | 2/28/1980 | 46 |
| | 0012350 | Putra | 8/1/1983 | 40 |
| * | | | | |

Misalnya Anda ingin mengubah nilai dari Danem menjadi 45 untuk data mahasiswa yang mempunyai NIM “0012348”, format penulisannya:

**Update MHS_3 Set Danem = 45 Where NIM = ‘001234'**

MHS_3 : Table

| | NIM | Nama | Tgl_Lahir | Danem |
|---|---|---|---|---|
| | 0012345 | Adinda | 3/17/1981 | 48 |
| ▶ | 0012348 | Kevin | 2/28/1980 | 45 |
| | 0012350 | Putra | 8/1/1983 | 40 |
| * | | | | |

Record: 2 of 3

Bila ingin mengubah lebih dari satu field maka format penulisannya:

**Update MHS_3 Set Nama = 'Putra Pratama' , Danem = Danem+5 Where NIM = '0012350'**

MHS_3 : Table

| NIM | Nama | Tgl_Lahir | Danem |
|---|---|---|---|
| 0012345 | Adinda | 3/17/1981 | 48 |
| 0012348 | Kevin | 2/28/1980 | 45 |
| 0012350 | Putra Pratama | 8/1/1983 | 45 |

Record: 3 of 3

**3) Menghapus Data**

Menghapus data adalah menghilangkan satu atau beberapa record data dari suatu tabel. Perintah query yang digunakan untuk menghapus adalah Delete, hanya dapat digunakan untuk menghapus record (baris) dan idak dapat digunakan untuk menghapus field (kolom). Untuk menentukan record yang akan dihapus dapat dilakukan perintah "Where". Jika tidak menggunakan perintah ini maka seluruh record yang ada pada tabel yang bersangkutan akan terhapus semua. Format penulisannya:

**DELETE [tabel.*] FROM tabel WHERE kriteria**

Keterangan :

| **Komponen** | **Keterangan** |
|---|---|
| Tabel.* | Optional nama tabel yang recordnya akan dihapus. |
| Tabel | Nama tabel dimana record-recordnya akan dihapus. |
| Kriteria | Ekspresi nilai baru sebagai pengganti Field1…FieldN. |
| Kriteria | Ekspresi criteria dari data yang akan dihapus. |

Sebagai contoh akan dihapus data record pada tabel MHS_3 yang Danem siswanya lebih kecil atau sama dengan 45.

**Delete * From MHS_3 Where Danem <= 45**

MHS_3 : Table

| NIM | Nama | Tgl_Lahir | Danem |
|---|---|---|---|
| 0012345 | Adinda | 3/17/1981 | 48 |

Record: 2 of 2

**c. Rangkuman**

Modifikasi data atau manipulasi data merupakan kegiatan yang bertujuan untuk mengubah data-data yang ada pada suatu tabel. Perubahan-perubahan data tersebut meliputi : manambah data, mengganti data maupun menghapus data . Menambah data baru kedalam suatu tabel dapat dilakukan dengan membuat tabel terlebih dahulu dimana data-data tersebut ditempatkan. Perintah yang digunakan untuk menambah data baru adalah insert into atau gabungan antara select dan insert into.

Pada kondisi tertentu kita ingin mengubah salah satu atau lebih field yang terdapat pada satu atua lebih record. Perintah yang digunakan adalah UPDATE. Menghapus data adalah menghilangkan satu atau beberapa record data dari suatu tabel. Perintah query yang digunakan untuk menghapus adalah Delete

**d. Tugas : mengoperasikan SQL untuk memodifikasi data.**

Dalam kegiatan ini peserta didik akan melakukan praktikum secara berkelompok, satu kelompok terdiri dari dua sampai tiga orang. Peserta didik akan menuliskan script SQL untuk memodifikasi data dalam database relasional. Bacalah seluruh langkah pengamatan dibawah ini kemudian lakukan dengan cermat dan teliti dengan perangkat yang telah disediakan.

1. Bentuk kelompok diskusi setiap kelompok terdiri dari dua-tiga orang.
2. Tulislah perintah-perintah SQL pada SQL view micorosoft acces untuk memasukkan atau menambahkan data pada setiap tabel yang telah dibuat pada kegiatan belajar 14. Untuk setiap tabel data yang ditambahkan minimal 4 record. Simpan query dan jalankan perintah SQL tersebut. Tulis dan catat hasilnya.
3. Dengan data yang telah dimasukkan tulislah perintah untuk SQL pada SQL View untuk memperbarui data, dan menghapus data. Simpan query dan jalankan perintah SQL tersebut. Tulis dan catat hasilnya.
4. Kumpulkan data-data setiap langkah dan analisis data tersebut menggunakan analisis diskriptif.

5. Komunikasikan hasilnya dalam kelompok, buatlah kesimpulan dan laporan sementara.
6. Komunikasikan dan verifikasi hasil kepada guru pembimbing.
7. Lampirkan hasil verifikasi dan pengesahan laporan sementara ke dalam laporan lengkap.

**e. Test Formatif.**

Dalam test ini setiap peserta didik membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

1. Sebutkan dan tuliskan contoh instruksi SQL untuk menambahkan sebuah data baru (satu record) dan lebih dari dua record.?
2. Sebutkan dan btuliskan contoh instruksi SQL untuk memperbarui dan menghapus data ?

**f. Lembar Jawaban Test Formatif (LJ).**

**LJ- 01 : Perintah SQL untuk menambahkan data baru (1 record dan lebih dari 2 record).**

.......................................................................................................
.......................................................................................................
.......................................................................................................
.......................................................................................................
.......................................................................................................
.......................................................................................................
.......................................................................................................
.......................................................................................................
.......................................................................................................
.......................................................................................................
.......................................................................................................
.......................................................................................................
.......................................................................................................
.......................................................................................................
.......................................................................................................
.......................................................................................................

**g. Lembar Kerja Peserta Didik.**

## Kegiatan belajar 16: Perintah SQL: Pengambilan Data

**a. Tujuan Pembelajaran.**

Setelah mengikuti kegiatan belajar 16 ini diharapkan peserta didik dapat:

- ✓ Mengoperasikan perintah SQL select beserta clausanya.
- ✓ Mengoperasikan perintah SQL dengan karakter khusus.
- ✓ Mengoperasikan perintah SQL dengan operator.
- ✓ Menggabungkan perintah SQL fungsi agregate.

**b. Uraian materi.**

Kegiatan yang paling sering dilakukan dalam pengoperasian database adalah mengambil data yang ada didalamnya. Kenyataan menunjukkan bahwa dalam satu aplikasi, operasi pengambilan data dari dalam database bisa mencapai lebih dari 80 % dari seluruh operasi yang berhubungan dengan database.

**1) Perintah Select untuk mengambil data**

Pemakaian perintah **Select** digunakan hanya untuk melakukan proses pengambilan data, bukan digunakan untuk mengubah data di dalam database. Pada pemakaian database relasional terdapat tiga macam operasi dasar terhadap suatu tabel yaitu **Selection, Projection** dan **Join**.

Selection didefinisikan sebagai pengambilan data secara horizontal pada suatu tabel yang memenuhi kondisi tertentu. Projection didefinisikan sebagai pengambilan data secara vertikal pada suatu tabel yang memenuhi baris-baris data yang unik. Join didefinisikan sebagai penggabungan satu tabel dengan tabel lainnya dalam ruang lingkup database relasional. Terdapat tiga jenis penggabungan yaitu **Inner Join, Left Join (left outer join),** dan **Right Join (right outer join)**.

Proses pengambilan data dari file database untuk ditampilkan ke layer dapat dilakukan dengan perintah SELECT. Perintah ini tidak menyebabkan data berubah atau terhapus sehingga bila sampai terjadi perubahan data , hal tersebut bukan lagi merupakan akibat dariperintah select.

Perintah select mempunyai beberapa anak kalimat yang dapat dikombinasikan dengan predikat dan kata cadangan yang telah disediakan oleh Microsoft Jet database Engine yang digunakan oleh Microsoft Access 2002.

keberadaan anak kalimat dan predikat dapat digunakan untuk menentukan data yang diambil, menyimpan data ke table sementara, mengurutkan data, mengelompokkan data dan lain-lain. Format penulisan pemakaian perintah Select adalah sebagai berikut :

**Select [predikat] { [tabel.]* | [ , [tabel.] Field1 [ , [tabel.] Field2 [ , …] ] }**
**[As alias1 [ , alias2 [ ,…] ] ]**
**From tabel [As tabel_alias] [ ,… [ As…]**
**[Where kritreria]**

Keterangan :

| Argumen | Keterangan |
|---|---|
| Predikat | Pilihan predikat yang akan digunakan. |
| Tabel | Nama tabel yang datanya akan diambil. |
| Field1, Field2, … | Nama-nama field yang datanya akan diambil. |
| Alias1, Alias2, … | Nama alias untuk field yang dispesifikasikan pada field. |
| Tabel_Alias | Nama alias untuk tabel yang dispesifikasikan. |
| Kriteria | Criteria pengambilan data. Dari seluruh record pada tabel, hanya record yang memenuhi kondisi pada anak kalimat Whereakan diambil. |

**2) Operasi Selection**

Operasi selection merupakan operasi untuk pengambilan data secara horizontal pada suatu tabel yang memenuhi kriteria tertentu. Sebagai conyoh akan digunakan tabel MHS_3 yang akan dikenakan operasi selection:

**Select * From MHS_3** atau **Select MHS_3.* From MHS_3**

MHS_3 : Table

| NIM | Nama | Tgl_Lahir | Danem |
|---|---|---|---|
| 0012345 | Adinda | 3/17/1981 | 45 |
| 003210 | Pitaloka | 9/23/1983 | 32 |
| 005678 | Kevin | 2/28/1983 | 43 |
| 982123 | Sekar | 9/15/1980 | 45 |
| 983213 | Manan | 7/29/1979 | 39 |
| 992345 | Haifa | 4/20/1982 | 45 |

Record: 1 of 6

Penerapan operasi selection yang diikuti oleh criteria yaitu "Menampilkan seluruh data dari tabel MHS_3 yang danemnya bernilai 45". Format penulisannya:

**Select * From MHS_3 Where Danem = 45** atau **Select MHS_3.* From MHS_3 Where MHS_3.Danem = 45**

querymhs : Select Query

| NIM | Nama | Tgl_Lahir | Danem |
|---|---|---|---|
| 0012345 | Adinda | 3/17/1981 | 45 |
| 982123 | Sekar | 9/15/1980 | 45 |
| 992345 | Haifa | 4/20/1982 | 45 |

Record: 1 of 3

### 3) Operasi Projection

Operasi Projection merupakan operasi untuk pengambilan data secara vertical pada suatu tabel yang mempunyai baris-baris data yang unik. Misal digunakan tabel MHS_3 untuk operasi projection.

**Select NIM, NAMA From MHS_3** atau **Select MHS_3.NIM, MHS_3.NAMA From MHS_3**

querymhs : Select Query

| NIM | NAMA |
|---|---|
| 0012345 | Adinda |
| 003210 | Pitaloka |
| 005678 | Kevin |
| 982123 | Sekar |
| 983213 | Manan |
| 992345 | Haifa |

Record: 1 of 6

Contoh pemakaian perintah Select unuk operasi selection dan projection, atas field-field NIM, NAMA dan DANEM dengan kriteria Danem=45.

**Select NIM, NAMA, DANEM From MHS_3 Where Danem=45** atau **Select MHS_3.NIM, MHS_3.NAMA, MHS_3.DANEM From MHS_3 Where MHS_3.Danem=45**

querymhs : Select Query

| NIM | NAMA | DANEM |
|---|---|---|
| 0012345 | Adinda | 45 |
| 982123 | Sekar | 45 |
| 992345 | Haifa | 45 |

Record: 1 of 3

#### 4) Ragam Predikat Pada Perintah Select

Perintah Select mempunyai beberapa predikat yaitu ALL, DISTINCT, DISTINCTROW dan TOP. Dengan mengikutsertakan predikat-predikat ini maka aturan penulisan select adalah sebagai berikut :

**Select [ALL | DISTINCTROW | [TOP n [PRECENT] ] ]**
**From Nama_Table**

#### a) Penggunaan ALL,

Jika Anda tidak menyertakan salah satu dari predikat yang telah disediakan, Microsoft Jet Engine akan menggunakan predikat ini dan akan memilih seluruh record yang memenuhi kondisi perintah SQL. Contoh penggunaan ALL :

**Select ALL * From MHS_3;** atau **Select * From MHS_3;**

querymhs : Select Query

| NIM | Nama | Tgl_Lahir | Danem |
|---|---|---|---|
| 0012345 | Adinda | 3/17/1981 | 45 |
| 003210 | Pitaloka | 9/23/1983 | 32 |
| 005678 | Kevin | 2/28/1983 | 43 |
| 982123 | Sekar | 9/15/1980 | 45 |
| 983213 | Manan | 7/29/1979 | 39 |
| 992345 | Haifa | 4/20/1982 | 45 |

Record: 1 of 6

#### b) Penggunaan DISTINC,

**D**engan perintah ini semua record yang isinya merupakan duplikasi field yang dipilih akan diabaikan. Data yang menggunakan predikat ini tidak dapat diubah dan refleksinya tidak dapat digunakan oleh user lain.

**Select DISTINC DANEM From MHS_3**

querymhs : Select Query

| DANEM |
|---|
| 32 |
| 39 |
| 43 |
| 45 |

Record: 1 of 4

#### c) Penggunaan DISTINCTROW,

Predikat ini hanya berfungsi jika Anda memilih beberapa field bukan seluruh field dari tabel yang digunakan perintah query. Predikat DISTINCTROW

akan mengabaikan fungsinya jika query yang dibuat hanya dikenakan pada satu tabel saja atau hasil tampilan fieldnya berasal dari seluruh tabel yang digunakan.

**Select DISTINCTROW DANEM From MHS_3**

querymhs : Select Query

| DANEM |
|---|
| 45 |
| 32 |
| 43 |
| 45 |
| 39 |
| 45 |

Record: 1 of 6

**d) Penggunaan TOP**

Digunakan untuk mengambil sejumlah record yang berada pada jangkauan atas atau bawah, dari seluruh data yang diperoleh. Argumen numeric yang diberikan pada predikat TOP defaultnya dalam bentuk jumlah record pada jangkauan atas yang akan ditampilkan. Sedangkan untuk mengambil n persen dari seluruh record yang didapat, gunakan argumen PERCENT.

**Select TOP 3 NIM, NAMA From MHS_3**

querymhs : Select Query

| NIM | NAMA |
|---|---|
| 0012345 | Adinda |
| 003210 | Pitaloka |
| 005678 | Kevin |

Record: 1 of 3

Contoh perintah sql menggunakan argumen PERCENT:

**Select TOP 20 PERCENT NIM, NAMA From MHS_3**

querymhs : Select Query

| NIM | NAMA |
|---|---|
| 0012345 | Adinda |
| 003210 | Pitaloka |

Record: 1 of 2

**5) Menentukan Kriteria Data**

Perintah yang merupakan anak kalimat SELECT dan digunakan untuk menentukan kriteria data adalah WHERE. Kreteria data yang akan diambil

meliputi menentukan data tertentu, mengabaikan data tertentu dan menjangkau data (interval) tertentu.

**a) Menentukan Data Tertentu**

Pengguna dapat menyertakan data-data tertentu pada hasil query dengan menggunakan beberapa operator Relasi. Operator-operator yang dimaksudkan adalah =, >, >=, < dan <=. Pemakaian operator ini dapat dikenakan untuk data string, number maupun date. Khusus untuk data string, jika nilai yang akan dibandingkan tidak dimasukkan ke dalam variable maka nilai tersebut harus ditulis dengan diapit oleh dua tanda petik (“”). Contoh perintah query untuk Menampilkan Data mahasiswa yang Mempunyai nilai [Danem] <=45

**Select * From MHS_3 Where DANEM <= 45**

**b) Mengabaikan Data Tertentu**

Kriteria dengan mengabaikan data adalah menampilkan data-data tertentu yang tidak diinginkan (tidak sesuai dengan criteria). Untuk melakukan proses seperti ini dapat digunakan operator <>. Perintah SQL dengan mengabaikan data tertentu adalah sebagai berikut:

**Select * From MHS_3 Where DANEM <> 45**

querymhs : Select Query

| NIM | Nama | Tgl_Lahir | Danem |
|---|---|---|---|
| 003210 | Pitaloka | 9/23/1983 | 32 |
| 005678 | Kevin | 2/28/1983 | 43 |
| 983213 | Manan | 7/29/1979 | 39 |

Record: 1 of 3

**c) Menjangkau Data Tertentu Dengan Perintah Between**

Perintah Select dapat juga digunakan untuk menampilkan data yang berada pada jangkauan criteria tertentu. Untuk itu operator yang digunakan adalah BETWEEN . . . END, yang memiliki format penulisan:

**Expr [NOT] BETWEEN Value1 AND Value2**

| **Argumen** | **Keterangan** |
|---|---|
| Expr | Ekspresi yg mengidentifikasi field yang berisi data yang akan dievaluasi. |

| Value1 | Ekspresi yang berisi nilai awal evaluasi. |
|---|---|
| Value2 | Ekspresi yang berisi nilai akhir evaluasi. |

**Catatan**:

- Apabila nilai expr antara valu1 dan value2 (inclusive), operator BETWEEN . . . END akan menghasilkan nilai TRUE dan sebaliknya.
- Jika expr, value1 atau value2 berisi NULL, operator BETWEEN . . . END akan menghasilkan NULL.
- Operator BETWEEN . . . END dapat digunakan pada ekspresi query dan kontrol kalkulasi pada form atau report.

Contoh Query untuk menampilkan data untuk Danem antara 35 dan 45

**Select * From MHS_3 Where DANEM BETWEEN 35 AND 45**

querymhs : Select Query

| NIM | Nama | Tgl_Lahir | Danem |
|---|---|---|---|
| 0012345 | Adinda | 3/17/1981 | 45 |
| 005678 | Kevin | 2/28/1983 | 43 |
| 982123 | Sekar | 9/15/1980 | 45 |
| 983213 | Manan | 7/29/1979 | 39 |
| 992345 | Haifa | 4/20/1982 | 45 |

Record: 1 of 5

Untuk melakukan evaluasi kembalikan suatu kondisi yaitu dengan menambahkan satu operator NOT.

**Select * From MHS_3 Where DANEM NOT BETWEEN 35 AND 45**

querymhs : Select Query

| NIM | Nama | Tgl_Lahir | Danem |
|---|---|---|---|
| 003210 | Pitaloka | 9/23/1983 | 32 |

Record: 1 of 1

Contoh query Menampilkan Data dengan Perintah Between untuk Tanggal

Select * from MHS_3 Where TGL_LAHIR BETWEEN #7/25/1979# AND #8/30/1981#

querymhs : Select Query

| NIM | Nama | Tgl_Lahir | Danem |
|---|---|---|---|
| 0012345 | Adinda | 3/17/1981 | 45 |
| 982123 | Sekar | 9/15/1980 | 45 |
| 983213 | Manan | 7/29/1979 | 39 |

Record: 1 of 3

Contoh query  menampilkan data yang namanya antara huruf A sampai R

**Select * From MHS_3 Where NAMA BETWEEN "A" AND "R"**

querymhs : Select Query

| NIM | Nama | Tgl_Lahir | Danem |
|---|---|---|---|
| 0012345 | Adinda | 3/17/1981 | 45 |
| 003210 | Pitaloka | 9/23/1983 | 32 |
| 005678 | Kevin | 2/28/1983 | 43 |
| 983213 | Manan | 7/29/1979 | 39 |
| 992345 | Haifa | 4/20/1982 | 45 |

Record: 1 of 5

**6) Menggunakan Kriteria Dengan Operator**

Operator logika (AND, OR, NOT, XOR) digunakan untuk menentukan kreteria dengan jumlah kondisi lebih dari satu pilihan kondisi. Peran operator adalah penghubung antara ekspresi satu dengan ekspresi lainnya.

**a) Operator AND**

Operator AND digunakan untuk menguji beberapa ekspresi logika yang diberikan memiliki nilai TRUE, dengan format penulisan:

**Ekspresi1 AND ekspresi2**

| **Ekspresi 1** | **Ekspresi 2** | **Hasil** |
|---|---|---|
| True | True | True |
| True | False | False |
| False | True | False |
| False | False | False |
| Null | True/False/Null | Null |
| True/False/Null | Null | Null |

Contoh query untuk menampilkan data mahasiswa yang berjenis kelamin "l" (laki-laki) dan bertempat tinggal dikota Surabaya :

**Select * From MHS_3 Where sex=l and kota=Surabaya**

querymhs : Select Query

| NIM | Nama | Tgl_Lahir | Danem | kota | sex |
|---|---|---|---|---|---|
| 005678 | Kevin | 2/28/1983 | 43 | surabaya | l |
| 983213 | Manan | 7/29/1979 | 39 | surabaya | l |

Record: 1 of 2

**b) Operator OR**

Operator OR digunakan untuk menguji apakah salah satu atau kedua ekspresi logika yang diberikan memiliki nilai TRUE, dengan format penulisan sebagai berikut :

**Ekspresi1 OR Ekspresi2**

Table kebenaran operator OR diperlihatkan dalamtabel dibawah ini

| Ekspresi 1 | Ekspresi 2 | Hasil |
|---|---|---|
| True | True | True |
| True | False | True |
| True | Null | True |
| False | True | True |
| False | False | False |
| False | Null | Null |
| Null | True | True |
| Null | False | False |
| Null | Null | Null |

Contoh query untuk menampilkan data mahasiswa yang berjenis kelamin “l”(lakilaki) atau bertempat tinggal di kota malang

**Select * From MHS_3 Where Sex=L OR Kota=Malang**

querymhs : Select Query

| NIM | Nama | Tgl_Lahir | Danem | kota | sex |
|---|---|---|---|---|---|
| 0012345 | Adinda | 9/15/1980 | 45 | malang | p |
| 005678 | Kevin | 2/28/1983 | 43 | surabaya | l |
| 983213 | Manan | 7/29/1979 | 39 | surabaya | l |

Record: 2 of 3

**c) Operator NOT**

Operator NOT digunakan untuk mendapatkan nilai kebaikan dari suatu logika atau ekspresi dengan format penulisannya: **NOT Ekspresi**.

| Ekspresi | Hasil |
|---|---|
| True | False |
| False | True |
| Null | null |

Contoh : perintah SQL untuk mendapatkan data mahasiswa yang mempunyai Danem > dari 40

**Select * From MHS_3 Where DANEM>40** atau
**Select * From MHS_3 Where NOT DANEM<=40**

querymhs : Select Query

| NIM | Nama | Tgl_Lahir | Danem | kota | sex |
|---|---|---|---|---|---|
| 0012345 | Adinda | 9/15/1980 | 45 | malang | p |
| 005678 | Kevin | 2/28/1983 | 43 | surabaya | l |
| 982123 | Sekar | 9/15/1980 | 45 | jakarta | p |
| 992345 | Haifa | 4/20/1982 | 45 | pasuruan | p |

Record: 3 of 4

### 7) Menguji Nilai Null

Data yang bernilai Null jika dibandingkan dengan data yang ada nilainya maka hasilnya akan Null, kondisi ini kurang valid jika ditinjau dari keberadaan data yang dipilih. Jika diinginkan untuk menyertakan record-record yang field-fieldnya diperkirakan ada yang berisi Null, sementara field-field tersebut digunakan pada ekspresi **Where** maka gunakan operator **IS NULL** sebagai tambahan kriteria untuk menyertakannya pada hasil query. Format penulisannya:

**ekspresi1 is [NOT] NULL**.

Contoh perintah SQL untuk menampilkan data dengan pengecekan NULL adalah sebagai berikut :

**Select NIM, NAMA, DANEM From MHS_3 Where DANEM=45 OR DANEM is Null**

querymhs : Select Query

| NIM | NAMA | DANEM |
|---|---|---|
| 0012345 | Adinda | 45 |
| 982123 | Sekar | 45 |
| 992345 | Haifa | 45 |

Record: 2 of 3

### 8) Menampilkan Ekspresi String Dengan Perintah Like

Dalam pencarian suatu ekspresi string dapat pula dilakukan secara variable dalam suatu ekspresi string lainnya, Microsoft Access 2002 menyediakan fasilitas query untuk melakukan proses tersebut yaitu dengan menggunakan operator **LIKE**. Dengan format penulisan: **ekspresi LIKE pola**.

| **Argumen** | **Keterangan** |
|---|---|
| Ekspresi | Ekspresi string dimana teks akan dicari. |
| Pola | Ekspresi string yang dapat berisi karakter wildchar (*, ?, #, !, , -, [ ]) dan teks yang akan dicari pada ekspresi. |

9) **Menggunakan karakter-karekter khusus**

a) **Menggunakan Karakter Wildchar Asterik**

Karakter asterik (*) digunakan untuk mengabaikan karakter apa saja setelah atau sebelum ekspresi string diletakkan. Jika karakter asterik diletakkan di depan suatu karater yang dicari maka akan terdapat satu atau beberapa karakter yang diabaikan sebelum karakter yang dicari. Sebaliknya jika karakter asterik diletakkan setelah karakter yang dicari maka akan terdapat satu atau beberapa karakter yang diabaikan setelah karakter yang dicari. Bila karakter asterik diletakkan di tengah antara dua atau lebih karakter yang dicari maka karakter yang di tengah tersebut akan diabaikan.

Contoh perintah SQL untuk mendapatkan data mahasiswa yang kota tinggalnya dimulai dengan karakter ”S”

**Select * from MHS_3 Where KOTA LIKE “S*”**

querymhs : Select Query

| NIM | Nama | Tgl_Lahir | Danem | kota | sex |
|---|---|---|---|---|---|
| 003210 | Pitaloka | 9/23/1983 | 32 | surabaya | p |
| 005678 | Kevin | 2/28/1983 | 43 | surabaya | l |
| 983213 | Manan | 7/29/1979 | 39 | surabaya | l |

Record: 3 of 3

Contoh perintah query untuk mendapatkan data mahasiswa yang kota tinggalnyadiakhiri dengan karakter “g”

**Select * From MHS_3 Where KOTA LIKE “*g”**

querymhs : Select Query

| NIM | Nama | Tgl_Lahir | Danem | kota | sex |
|---|---|---|---|---|---|
| 0012345 | Adinda | 9/15/1980 | 45 | malang | p |

Record: 1 of 1

Gambar 4.20 Parameter LIKE dengan Karakter * di Depan

Contoh perintah SQL untuk mendapatkan data mahasiswa yang kota tinggalnya terdapat karakter ”raba”

**Select * From MHS_3 Where KOTA LIKE “*raba*”**

querymhs : Select Query

| NIM | Nama | Tgl_Lahir | Danem | kota | sex |
|---|---|---|---|---|---|
| 003210 | Pitaloka | 9/23/1983 | 32 | surabaya | p |
| 005678 | Kevin | 2/28/1983 | 43 | surabaya | l |
| 983213 | Manan | 7/29/1979 | 39 | surabaya | l |

Record: 4 of 4

**b) Menggunakan Karakter Tanda Tanya**

Karakter tanda Tanya digunakan untuk menggantikan (mengabaikan) satu karakter pada posisi yang diinginkan. Contoh : Dapatkan data mahasiswa yang namanya diawali dengan huruf ”P” dan pada karakter ke tiga terdapat huruf k

**Select * From MHS_3 Where NAMA LIKE “S?k*”**

QUERYJENIS : Select Query

| NIM | Nama | Tgl_Lahir | Danem | kota | sex |
|---|---|---|---|---|---|
| 982123 | Sekar | 9/15/1980 | 45 | jakarta | p |
| 645923 | Sukawati | 9/16/1982 | 46 | malang | p |

Record: 2 of 2

**c) Menggunakan Karakter Bracket**

Karakter Bracket ([ ]) digunakan untuk menunjukan bahwa satu karakter yang diganti (dipilih) harus berada pada daftar. Sebagai contoh akan ditampilkan data mahasiswa dimana namanya pada karakter kedua berisi huruf a atau e, maka format penulisannya:

**select * from MHS_3 Where NAMA LIKE “?[ae]*”**

QUERYJENIS : Select Query

| NIM | Nama | Tgl_Lahir | Danem | kota | sex |
|---|---|---|---|---|---|
| 005678 | Kevin | 2/28/1983 | 43 | surabay | l |
| 982123 | Sekar | 9/15/1980 | 45 | jakarta | p |
| 983213 | Manan | 7/29/1979 | 39 | surabay | l |
| 992345 | Haifa | 4/20/1982 | 45 | pasurua | p |

Record: 2 of 4

**d) Menggunakan Karakter Tanda Seru**

Karakter tanda seru (!) digunakan untuk menunjukkan bahwa karakter yang diganti (dipilih) hanya untuk karakter yang tidak terdapat pada daftar. Sebagai contoh akan tampilkan data mahasiswa namanya pada karakter ke dua bukan huruf u, maka format penulisannya:

**Select * From MHS_3 Where NAMALIKE “?[!u]*”**

QUERYJENIS : Select Query

| NIM | Nama | Tgl_Lahir | Danem | kota | sex |
|---|---|---|---|---|---|
| 0012345 | Adinda | 9/15/1980 | 45 | malang | p |
| 003210 | Pitaloka | 9/23/1983 | 32 | surabaya | p |
| 005678 | Kevin | 2/28/1983 | 43 | surabaya | l |
| 982123 | Sekar | 9/15/1980 | 45 | jakarta | p |
| 983213 | Manan | 7/29/1979 | 39 | surabaya | l |
| 992345 | Haifa | 4/20/1982 | 45 | pasuruan | p |

Record: 5 of 6

**e) Menggunakan Karakter Tanda Minus**

Karakter tanda minus (-) digunakan untuk mengganti satu karakter yang berada pada jangkauan tertentu. Missal akan ditampilkan data yang namanya pada karakter kedua mulai dari huruf “a” sampai “g”.`Format penulisannya:

**Select * From MHS_3 Where NAMA LIKE “?[a-g]*”**

QUERYJENIS : Select Query

| NIM | Nama | Tgl_Lahir | Danem | kota | sex |
|---|---|---|---|---|---|
| 0012345 | Adinda | 9/15/1980 | 45 | malang | p |
| 005678 | Kevin | 2/28/1983 | 43 | surabaya | l |
| 982123 | Sekar | 9/15/1980 | 45 | jakarta | p |
| 983213 | Manan | 7/29/1979 | 39 | surabaya | l |
| 992345 | Haifa | 4/20/1982 | 45 | pasuruan | p |

Record: 1 of 5

**f) Menggunakan Karakter Simbol Nomor**

Karakter tanda simbol nomor (#) digunakan untuk mengganti satu digit angka, jadi karakter ini tidak dapat digunakan selain number. Missal akan ditampilkan data yang memiliki danem dengan digit pertama berangka 4, format penulisannya:

**Select * From MHS_3 Where DANEM LIKE “4#”**

QUERYJENIS : Select Query

| NIM | Nama | Tgl_Lahir | Danem | kota | sex |
|---|---|---|---|---|---|
| 0012345 | Adinda | 9/15/1980 | 45 | malang | p |
| 005678 | Kevin | 2/28/1983 | 43 | surabaya | l |
| 982123 | Sekar | 9/15/1980 | 45 | jakarta | p |
| 992345 | Haifa | 4/20/1982 | 45 | pasuruan | p |
| 645923 | Sukawati | 9/16/1982 | 46 | malang | p |

Record: 1 of 5

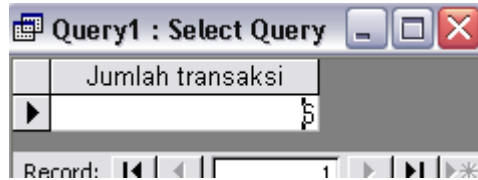
**10) Menggunakan Fungsi *Agregate* Pada Ekspresi Where**

Fungsi anggregate adalah fungsi yang digunakan untuk melakukan operasi terhadap himpunan data dalam perintah query. Fungsi *agregate* yang disediakan oleh Microsoft Access adalah SUM, COUNT, AVG, MIN, MAX.

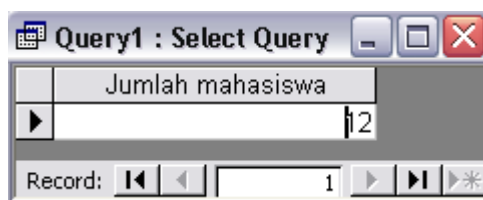
**a) Fungsi SUM**

Fungsi sum mempunyai bentuk SUM(x) digunakan untuk menghasilkan nilai penjumlahan dari suatu data bilangan dimana x adalah nilai numerik atau nama field yang memiliki nilai numeric.

Contoh perintah SQL untuk mendapatkan data jumlah transaksi dari table jual

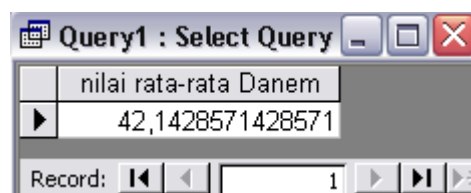
```
SELECT  SUM(JML_TRANS)  as [Jumlah transaksi]
FROM JUAL;
```

## b) Fungsi COUNT

Fungsi Count yang mempunyai bentuk COUNT(field) digunakan untuk menghitung banyaknya data dalam suatu table dari hasil query. Dengan fungsi ini dapat dihitung jumlah data yang diperoleh atas dasar field tertentu atau seluruh field pada query. Contoh perintah SQL untuk mendapatkan data jumlah mahasiswa

```
SELECT COUNT(NIM_MHS)  AS [Jumlah mahasiswa]
FROM DAFTAR_MHS;
```

## c) Fungsi AVG

Fungsi AVG yang mempunyai bentuk AVG(x) digunakan untuk menghasilkan nilai rata-rata dari suatu data bilangan. Dimana x adalah nilai numerik atau nama field yang memiliki nilai numerik atau rumus yang menghasilkan nilai numerik

Contoh : dapatkan nilai rata-rata danem dari daftar mahasiswa di bawah ini

```
SELECT AVG(Danem)  as [nilai rata-rata Danem]
FROM  MHS_3;
```

**11) Mengurutkan data.**

Proses Mengurutkan data (Sort) bertujuan untkmenata data-data yang ada agar urutan tampilan nya sesuai dengan yang diinginkan. Pola pengurutan yang dilakukan ada dua macam, yaitu :

1. Ascending : ialah mengurutkan data mulai dari data terkecil sampai data terbesar (menaik). Untuk data string (teks) maka urutan dimulai dai abjac "A" samapai "Z". Sedangkan untukdata number (numerik) dimulai dari angka 0 sampai 9
2. DesCending : ialah mengurutkan data mulai dari data terbesar sampai data terkecil (menurun). Untuk data string (teks) maka urutan dimulai dai abjac "Z" samapai "A". Sedangkan untuk data number (numerik) dimulai dari angka 9 sampai 0

Proses pengurutan data dengan perintah query dapat dilakukan dengan menggunakan anak kalimat ORDER BY yang merupakan bagian dari perintah SELECT. Sedara default proses pengurutan data pada kata tercadang ORDER BY adalah Ascending. Aturan penulisan sebagai berikut :

**SELECT ......**
**FROM ......**
**WHERE ....**
**ORDER BY [Tabel.]Field1 [ASC | DESC] [, [Tabel.]Field2 [ASC | DESC]**

**Keterangan:**

| **Argumen** | **Keterangan** |
|---|---|
| Table.Field1 | Field1 yang digunakan sebagai kunci pertama pengurutan |
| Table.Field2 | Field2 yang digunakan sebagai kunci kedua pengurutan |
| ASC \| DESC | Pola pengurutan yaitu ascending atau descending |

Contoh perintah SQL untuk mengurutkan data barang berdasarkan nama secara descending

**SELECT * FROM barang ORDER BY Nama DESC**

Contoh perintah SQL untuk mengurutkan data barang dengan kunci pertama field jumlah secara ascending dan kunci ke dua field Harga secara Descending berdasarkan nama secara descending

**SELECT * FROM barang ORDER BY Jumlah, Harga DESC**

**12) Mengelompokkan Data**

Proses pengelompokan data (record) yang sama dengan perintah query dapat dilakukan dengan menggunakan anak kalimat GROUP BY yang merupakan bagian dari perintah SELECT. Pada kasus yang sederhana anak kalimat GROUP BY berfungsi seperti perintah DISTICT. Kelebihan dari anak kalimat GROUP BY adalah dapat digunakan bersamadengan fungsi anggregate. Fungsi anggregate yang digunakan bersama GROUP BY akan beroperasi pada seluruh record yang akan digabung untuk membentuk satu record (baris) tunggal.

Banyak hal yang dapat diperolehb dari proses pengelompokan data, yaitu untukmelakukan perhitungan akumulasi suatu nilai pada setiap kelompok, mendapatkan nilai rata-rata, nilai maksimum,minimum, dansebagainya. Aturan penulisan nya adalah sebagai berikut :

**SELECT ......**
**FROM ......**
**WHERE ....**
**ORDER BY ……**
**GROUP BY [table.]Field1 [, [table.]Field2 [,…..]**
**HAVING Kriteria Group]**

| Argumen | Keterangan |
|---|---|
| Table.Field1 | Field1 yang digunakan sebagai kunci pertama pengelompokan |
| Table.Field2 | Field2 yang digunakan sebagai kunci kedua pengelompokan |
| ASC \| DESC | Mengevaluasi kondisi (kriteria) |

**c. Rangkuman**

Kegiatan yang paling sering dilakukan dalam pengoperasian database adalah proses pengambilan data. Operasi pengambilan data dari dalam database bisa mencapai lebih dari 80 % dari seluruh operasi yang berhubungan dengan database. Pada pemakaian database relasional terdapat tiga macam operasi dasar terhadap suatu tabel yaitu Selection, Projection dan Join. Selection didefinisikan sebagai pengambilan data secara horizontal pada suatu tabel yang memenuhi kondisi tertentu. Projection didefinisikan sebagai pengambilan data secara vertikal pada suatu tabel yang memenuhi baris-baris data yang unik. Join

didefinisikan sebagai penggabungan satu tabel dengan tabel lainnya dalam ruang lingkup database relasional

Perintah Select mempunyai beberapa predikat yaitu ALL, DISTINCT, DISTINCTROW dan TOP. Perintah yang merupakan anak kalimat SELECT dan digunakan untuk menentukan kriteria data adalah WHERE. Operator logika (AND, OR, NOT, XOR) digunakan untuk menentukan kreteria dengan jumlah kondisi lebih dari satu pilihan kondisi. Peran operator adalah penghubung antara ekspresi satu dengan ekspresi lainnya.

Microsoft Access menyediakan fasilitas query untuk menampilkan string dengan menggunakan operator LIKE. Microsoft access juga menyediakan beberapa karakter-karakter khusus untuk ekspresi string seperti Karakter asterik (*), tanda Tanya, Bracket ([ ]) tanda seru (!), tanda minus (-) dan tanda simbol nomor (#). Fungsi anggregate adalah fungsi yang digunakan untuk melakukan operasi terhadap himpunan data  dalam perintah query. Fungsi anggregate yang disediakan oleh Microsoft Access  adalah SUM, COUNT, AVG, MIN, MAX.

**d. Tugas : mengoperasikan SQL untuk mengambil data.**

Dalam kegiatan ini peserta didik akan melakukan praktikum secara berkelompok, satu kelompok terdiri dari dua sampai tiga orang. Peserta didik akan menuliskan script SQL untuk mengambil data dalam database relasional. Bacalah seluruh langkah pengamatan dibawah ini kemudian lakukan dengan cermat dan teliti dengan perangkat yang telah disediakan.

1. Bentuk kelompok diskusi setiap kelompok terdiri dari dua-tiga orang.
2. Dengan menggunakan schema data base pada kegiatan belajar 14, tulislah perintah-perintah SQL pada SQL view micorosoft acces untuk proses pengambilan data menggunakan perintah *selection, projection* dan *joint*. Simpan query dan jalankan perintah SQL tersebut. Tulis dan catat hasilnya.
3. Ulangi langkah 2 untuk penyeleksian data dengan melibatkan beberapa operator. Simpan query dan jalankan perintah SQL tersebut. Tulis dan catat hasilnya.

4. Ulangi langkah 2 untuk penyeleksian dan pengambilan data dengan melibatkan beberapa karakter khusus untuk ekspresi string. Simpan query dan jalankan perintah SQL tersebut. Tulis dan catat hasilnya.
5. Ulangi langkah 2 untuk penyeleksian dan pengambilan data dengan melibatkan beberapa fungsi *agregate.* Simpan query dan jalankan perintah SQL tersebut. Tulis dan catat hasilnya
6. Kumpulkan data-data setiap langkah dan analisis data tersebut menggunakan analisis diskriptif.
7. Komunikasikan hasilnya dalam kelompok, buatlah kesimpulan dan laporan sementara.
8. Komunikasikan dan verifikasi hasil kepada guru pembimbing.
9. Lampirkan hasil verifikasi dan pengesahan laporan sementara ke dalam laporan lengkap.

**e. Test Formatif.**

Dalam test ini setiap peserta didik membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

1. Tuliskan, jelaskan dan berikan contoh format penulisan SQL menggunakan perintah *select* beserta *clausa-clausanya* .?
2. Sebutkan dan jelaskan (kegunaannya) ragam karakter khusus untuk penyeleksian dan pengambilan data string pada microsoft access ?
3. Sebutkan dan jelaskan (kegunaannya) ragam fungsi agregate pada microsoft acces ?

**f. Lembar Jawaban Test Formatif (LJ).**

**LJ- 01 : Format penulisan SQL perintah SELECT dan Clausanya .**

..........................................................................................................

..........................................................................................................

..........................................................................................................

..........................................................................................................

..........................................................................................................

..........................................................................................................

g. **Lembar Kerja Peserta Didik.**

**Kegiatan belajar 17: Sistem manajemen basis data**

**a. Tujuan Pembelajaran.**

Setelah mengikuti kegiatan belajar 17 ini diharapkan peserta didik dapat:

- ✓ Memahami Konsep sistem manajemen basis data (SMBD)
- ✓ Memahami Sistem manajemen basis data relasional (RDBMS)
- ✓ Membedakan berbagai jenis SMBD

**b. Uraian materi.**

**1) Definisi Sistem manajemen basis Data**

Sistem manajemen basis data (*database management system*, DBMS), atau sering disingkat SMBD, adalah suatu sistem atau perangkat lunak yang dirancang untuk mengelola suatu basis data dan menjalankan operasi terhadap data yang diminta banyak pengguna. Contoh tipikal SMBD adalah akuntansi, sumber daya manusia, dan sistem pendukung pelanggan. SMBD telah berkembang menjadi bagian standar di bagian pendukung (*back office*) suatu perusahaan atau organisasi. Contoh SMBD adalah Oracle, SQL server 2000/2003, MS Access, MySQL dan sebagainya.

DBMS merupakan perangkat lunak yang dirancang untuk dapat melakukan utilisasi dan mengelola koleksi data dalam jumlah yang besar. DBMS juga dirancang untuk dapat melakukan manipulasi data secara lebih mudah. Sebelum ada DBMS, data pada umumnya disimpan dalam bentuk flat file, yaitu file teks yang ada pada sistem operasi. Sampai sekarangpun masih ada aplikasi yang menimpan data dalam bentuk flat secara langsung. Menyimpan data dalam bentuk flat file mempunyai kelebihan dan kekurangan. Penyimpanan dalam bentuk ini akan mempunyai manfaat yang optimal jika ukuran filenya relatif kecil, seperti file passwd pada sistem operasi Unix dan Unix-like. File passwd pada umumnya hanya digunakan untuk menyimpan nama yang jumlahnya tidak lebih dari 1000 orang.

Selain dalam bentuk flat file, penyimpanan data juga dapat dilakukan dengan menggunakan program bantu seperti *spreadsheet*. Penggunaan perangkat lunak ini memperbaiki beberapa kelemahan dari flat file, seperti bertambahnya kecepatan dalam pengolahan data. Namun demikian metode ini

masih memiliki banyak kelemahan, diantaranya adalah masalah manajemen dan keamanan data yang masih kurang. Penyimpanan data dalam bentuk DBMS mempunyai banyak manfaat dan kelebihan dibandingkan dengan penyimpanan dalam bentuk flat file atau spreadsheet, diantaranya :

1. Performa yang dapat dengan penyimpanan dalam bentuk DBMS cukup besar, sangat jauh berbeda dengan performance data yang disimpan dalam bentuk flat file. Disamping memiliki unjuk kerja yang lebih baik, juga akan didapatkan efisiensi penggunaan media penyimpanan dan memori
2. Integritas data lebih terjamin dengan penggunaan DBMS. Masalah redudansi sering terjadi dalam flat file. Redudansi adalah kejadian berulangnya data atau kumpulan data yang sama dalam sebuah database yang mengakibatkan pemborosan media penyimpanan.
3. Independensi. Perubahan struktur database dimungkinkan terjadi tanpa harus mengubah aplikasi yang mengaksesnya sehingga pembuatan antarmuka ke dalam data akan lebih mudah dengan penggunaan DBMS.
4. Sentralisasi. Data yang terpusat akan mempermudah pengelolaan database. Kekonsistenan data yang diakses secara bersama-sama akan lebiih terjamin dari pada data disimpan dalam bentuk file atau worksheet yang tersebar.
5. Keamanan. DBMS memiliki sistem keamanan yang lebih fleksibel daripada pengamanan pada file sistem operasi. Keamanan dalam DBMS akan memberikan keluwesan dalam pemberian hak akses kepada pengguna.

**2) Sistem manajemen basis data relasional**

Sebuah sistem manajemen basis data relasional atau dikenal sebagai *relational database management system (RDBMS)* adalah sebuah program komputer (atau secara lebih tipikal adalah seperangkat program komputer) yang dirancang untuk mengatur atau mengelola sebuah basis data sebagai sekumpulan data yang disimpan secara terstruktur, dan melakukan operasi-operasi data atas permintaan penggunanya. Contoh penggunaan DBMS ada banyak sekali dan dalam berbagai bidang kerja, misalnya akuntansi, manajemen sumber daya manusia, dan lain sebagainya. Meskipun pada awalnya DBMS hanya dimiliki oleh perusahaan-perusahaan berskala besar yang memiliki

perangkat komputer yang sesuai dengan spesifikasi standar yang dibutuhkan (pada saat itu standar yang diminta dapat dikatakan sangat tinggi) untuk mendukung jumlah data yang besar, saat ini implementasinya sudah sangat banyak dan adaptatif dengan kebutuhan spesifikasi data yang rasional sehinggal dapat dimiliki dan diimplementasikan oleh segala kalangan sebagai bagian dari investasi perusahaan.

Edgar F. Codd memperkenalkan istilah RDBMSi pada makalah seminarnya yang berjudul "A Relational Model of Data for Large Shared Data Banks". Salah satu definisi yang cukup dikenal secara luas atas sebuah sistem basis data relasional adalah 12 hukum Codd. Namun demikian, pada awal implementasinya banyak model relasional yang tidak mengikuti seluruh elemen-elemen yang terdapat dalam hukum-hukum Codd tersebut yang menjadikan terminologinya berkembang untuk mendeskripsikan sebuah tipikal sistem basis data yang lebih luas. Dalam cakupan yang minimum sistem tersebut memenuhi kriteria berikut:

- menyajikan data pada pengguna dalam bentuk relasional (ditampilkan dalam bentuk tabular, sebagai koleksi dari tabel dimana setiap tabel beriisi sekumpulan baris dan kolom)
- menyediakan operator relasional untuk memanipulasi data dalam bentuk tabular

Sistem yang pertama kalinya yang secara relatif memenuhi implementasi atas sebuah model relasional adalah Pusat Studi Ilmiah IB, Inggris, di Peterlee; IS1 (1970-1972) dan implementasi lain yang mengikutinya PRTV (1973-1979). Sistem yang pertama kalinya dijual secara komersil sebagai RDBMS adalah Multics Relational Data Srore pada tahun 1978. Yang lainnya adalah Berkeley Ingres QUEL dan IBM BS12.

**3) *Hukum Codd***

Hukum cood adalah suatu ketentuan atau aturan dan definisi standar dari sebuah sistem basis data relasional, yang diperkenalkan oleh Edgar F. Codd. Hukum Codd terdiri dari dua belas kriteria atau ketentuan yaitu :

1. **Hukum 0:** Suatu sistem harus memenuhi kualifikasi sebagai *relasional*, sebagai *basisdata*, dan sebagai sebuah *sistem manajemen*:

2. **Hukum 1:** Hukum informasi: Seluruh informasi yang terdapat dalam basisdata harus bisa direpresentasikan hanya dalam satu cara, yaitu dalam bentuk nilai-nilai yang terisi dalam bentuk tabular baris dan kolom.
3. **Hukum 2:** Hukum Jaminan akses:Seluruh data harus bisa diakses tanpa ada kerancuan (ambiguity). Hukum ini merupakan penegasan dari kebutuhan mendasar atas sebuah kunci primer. Hukum tersebut menjelaskan bahwa setiap nilai skalar dalam basisdata haruslah memiliki alamat secara logikal dengan cara menspesifikasikan nama dari tabel, nama dari kolom, dan nilai kunci primer dari baris data dalam tabel tersebut.
4. **Hukum 3:** Perlakuan sistematik terhadi nilai NULL:Sebuah sistem manajemen basisdata harus mengijinkan setiap field terisi dengan nilai NULL (kosong). Sistem harus mendukung representasi dari "Hilangnya informasi dan Ketidakbergunaan informasi" secara sistematis, membedakan secara jelas dari nilai-nilai yang lain (contoh: "perbedaan antara nol dengan nilai-nilai numerik lain," dalam kasus nilai-nilai numerik), dan tipe data yang bersifat independen. Termasuk pula representasi tersebut harus dapat dimanipulasi oleh DBMS melalui langkah-langkah yang sistematis.
5. **Hukum 4:** Katalog online yang aktif haruslah berbasis model relasional: Sistem harus mendukung sebuah katalog relasional yang bersifat online, inline yang bisa diakses untuk pengguna yang sah dalam arti melalui bahasa kueri reguler. Lebih jelas lagi, pengguna harus dapat mengakses struktur data tersebut (katalog) dengan cara yang sama menggunakan bahasa kueri yang digunakan pula untuk mengakses data.
6. **Hukum 5:** Hukum sub-bahasa data yang komprehensif: Sistem harus mendukung setidaknya satu bahasa relasional yang
   a. Memiliki sintaksis linear
   b. Dapat digunakan secara interaktif maupun melalui program aplikasi
   c. Mendukung operasi pendefinisian data (termasuk pendefinisan view), operasi manipulasi data, aspek kemanan dan pembatasan integritas, operasi-operasi manajemen transaksi (begin, commit, dan rollback).

7. **Hukum 6:** Hukum pembaruan/update view:Semua view yang secara teoritis dapat diupdate dalam implementasinya juga harus dapat diupdate oleh sistem.
8. **Hukum 7:** Level tingkat tinggi dalam operasi insert, update, dan delete: Sistem harus mendukung serangkaian operasi-operasi *insert*, *update*, and *delete* dalam satu masa waktu yang sama.
9. **Hukum 8:** Data secara fisik bersifat independen: Perubahan pada level fisik (bagaimana suatu data disimpan, menggunakan larik ataupun senarai berantai dsb.) harus tidak mengakibatkan perubahan pada struktur di sisi aplikasi pada level yang lebih tinggi.
10. **Hukum 9:** Data secara logikal bersifat independen: Perubahan pada level logikal (Tabel, kolom, baris) harus tidak mengakibatkan perubahan pada level struktur di sisi aplikasi pada level ang lebih tinggi. Hukum ini secara relatif lebih suli dicapai daripada hukum 8.
11. **Hukum 10:** Integritas data bersifat independen: Integritas data harus dispesifikasikan secara terpisah dari program aplikasi dan disimpan dalam katalog/struktur dan harus memungkinkan untuk melakukan perubahan terhadap struktur tersebut ketika dibutuhkan tanpa memengaruhi aplikasi yang telah ada.
12. **Hukum 11:** Distribusi yang bersifat independen: Distribusi atas sebagian dari basisdata ke berbagai lokasi harus dapat diatur sedemikian rupa sehingga tidak terlihat oleh pengguna dari basisdata tersebut. Begitu pula aplikasi-aplikasi yang ada harus tetap dapat beroperasi secara normal seperti biasanya ketika:
    a. saat versi dari DBMS yang terdistribusi pertamakali diperkenalkan; dan/ataupun
    **b.** ketika data-data yang terdistribusi tersebut diredistribusikan ke seluruh sistem**.**
13. **Hukum 12:** Hukum nonsubversion: Jika sebuah sistem menyediakan antarmuka tingkat rendah, maka antarmuka tersebut tidak dapat digunakan untuk menggagalkan sistem, sebagai contoh, membypass aturan-aturan yang terkait dengan keamanan data, ataupun integritasnya.

**4) Ragam jenis SMBD**

Beberapa software atau perangkat lunak DBMS yang sering digunakan dalam membuat aplikasi program basis data dan sangat populer antara lain adalah MySQL, MS SQL Server, Oracle, IBM DB/2, dan PostgreSQL. Sementara di pasaran terdapat sejumlah software DBMS baik yang komersial atau open source antara lain adalah sebagai berikut: :

a. DBMS yang bersifat komersial: 4th Dimension, Dataphor, Daffodil database, DB2, FileMaker Pro, FrontBase, Informix, InterBase, Matisse [1], Microsoft Access, Microsoft SQL Server, Microsoft Visual FoxPro, Mimer SQL, Netezza, NonStop SQL, Oracle, Progress 4GL, Sand Analytic Server (sebelumnya dikenal sebagai Nucleus), SmallSQL, Sybase Adaptive Server Anywhere (sebelumnya dikenal sebagai Watcom SQL), Sybase Adaptive Server Enterprise, Sybase Adaptive Server IQ, Teradata, ThinkSQL [2], VistaDB, VMDS.

b. DBMS yang bersifat open source: antara lain : Cloudscape, Derby, Firebird, H2, HSQLDB, Ingres, MaxDB, MonetDB, MySQL, PostgreSQL, SQLite, tdbengine

**5) MySQL**

MySQL adalah sebuah perangkat lunak sistem manajemen basisdata relasional (RDBMS) yang mendukung sistem multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public Licenci (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL . Tidak seperti Apache yang merupakan software yang dikembangkan oleh komunitas umum, dan cipta untuk code sumber dimiliki oleh penulisnya masing-masing.

MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang

mendirikan MySQL AB adalah : david axmark, allan larsson, dan Michael “monthy widenius. MySQL memiliki beberapa kelebihan dan keistimewaan antara lain adalah sebagai berikut :

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. Multi-user. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. 'Performance tuning', MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti signed/unsigned integer, float, double, char, text, date, timestamp,
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (*query*).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket(UNIX), Named Pipes (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (tool)yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

**6) *Oracle***

Oracle adalah relational database management system (RDBMS) untuk mengelola informasi secara terbuka, komprehensif dan terintegrasi. RDBMS Oracle pertama kali dikembangkan oleh Larry Ellison, Bob Miner dan Ed Oates lewat perusahaan konsultasinya bernama *Software Development Laboratories (SDL)* pada tahun 1977. Pada tahun 1983, perusahaan ini berubah nama menjadi Oracle Corporation. Oracle Server menyediakan solusi yang efisien dan efektif karena kemampuannya dalam hal sebagai berikut:

- Dapat bekerja di lingkungan client/server (pemrosesan tersebar)
- Menangani manajemen space dan basis data yang besar
- Mendukung akses data secara simultan
- Performansi pemrosesan transaksi yang tinggi
- Menjamin ketersediaan yang terkontrol
- Lingkungan yang terreplikasi

Oracle merupakan DBMS yang paling rumit dan paling mahal di dunia, namun banyak orang memiliki kesan yang negatif terhadap Oracle. Keluhan-keluhan yang mereka lontarkan mengenai Oracle antara lain adalah

terlalu sulit untuk digunakan, terlalu lambat, terlalu mahal. Jika dibandingkan dengan MySQL yang bersifat gratis, maka Oracle lebih terlihat tidak kompetitif karena berjalan lebih lambat daripada MySQL meskipun harganya sangat mahal.

Oracle merupakan DBMS yang dirancang khusus untuk organisasi berukuran besar, bukan untuk ukuran kecil dan menengah. Kebutuhan organisasi berukuran besar tidaklah sama dengan organisasi yang kecil atau menengah yang tidak akan berkembang menjadi besar. Organisasi yang berukuran besar membutuhkan fleksibilitas dan skalabilitas agar dapat

memenuhi tuntutan akan data dan informasi yang bervolume besar dan terus menerus bertambah besar.

Kelebihan oracle adalah fleksibilitas sistem yaitu kemampuan untuk menyesuaikan diri dengan berbagai kebutuhan dan kondisi khusus yang dapat berubah-ubah. Sebagai contoh, organisasi yang besar membutuhkan server yang terdistribusi dan memiliki redundancy sehingga pelayanan bisa diberikan secara cepat dan tidak terganggu jika ada server yang mati. Organisasi tersebut juga mempunyai berbagai macam aplikasi yang dibuat dengan beragam bahasa pemrograman dan berjalan di berbagai platform yang berbeda. Oracle memiliki banyak sekali fitur yang dapat memenuhi tuntutan fleksibilitas dari organisasi besar tersebut. Berbagai fitur tersebut membuat Oracle menjadi DBMS yang rumit dan sulit untuk dipelajari, namun itu adalah harga yang harus dibayar untuk mendapatkan fleksibilitas yang dibutuhkan dalam sistem informasi di organisasi yang berukuran besar.

Kelebihan lainnya adalah skalabilitas yang mengacu pada kemampuan untuk terus berkembang dengan penambahan sumber daya. Organisasi yang besar harus mampu melakukan transaksi data dalam volume yang besar dan akan terus bertambah besar. Jika dijalankan hanya pada satu server saja, MySQL memang bisa berjalan lebih cepat daripada Oracle. Namun jika satu server sudah tidak bisa lagi menangani beban yang terus bertambah besar, kinerja MySQL mengalami stagnasi karena keterbatasan server tersebut. Namun Oracle mendukung fitur Grid yang dapat mendayagunakan lebih dari satu server serta data storage dengan mudah dan transparan. Hanya dengan menambahkan server atau data storage ke dalam Oracle Grid, maka kinerja dan kapasitas Oracle dapat terus berkembang untuk mengikuti beban kerja yang terus meningkat.

.

**7) FireBird**

Firebird adalah salah satu aplikasi RDBMS (Relational Database Management System) yang bersifat open source. Awalnya perusahaan Borland pada tahun 2000 mengeluarkan versi beta dari aplikasi yaitu InterBase 6.0 dengan sifat open source. Pengembangan basis data ini diawali dari Firebird 1, dengan merubah kode berbasis C ke dalam bahasa C++ dan merupakan

pembersihan kode secara besar-besaran. Firebird 1.5 merupakan rilis pertama dari codebase Firebird 2.

Firebird (juga disebut FirebirdSQL) adalah sistem manajemen basisdata relasional yang menawarkan fitur-fitur yang terdapat dalam standar ANSI SQL-99 dan SQL-2003. RDBMS ini berjalan baik di Linux, Windows, maupun pada sejumlah platform Unix. Firebird ini dikembangkan oleh FirebirdSQL Foundation. DBMS ini merupakan turunan dari Interbase versi open source milik Borland. Beberapa kelebihan open source DBMS ini antara lain:

1. Firebird support dengan transaksi layaknya pada database komersial lainnya. Sebuah transaksi bisa di-commit atau di-rollback dengan mudah. Firebird support dengan savepoint pada transaksi dan bisa melakukan rollback kembali ke savepoint (fungsi ini seperti fasilitas pada Oracle).
2. Firebird menggunakan sintaks standard untuk menciptakan suatu foreign key.
3. Firebird support row level locks, secara default Firebird menggunakan apa yang disebut dengan multi-version
4. Concurrency system. Ini artinya bahwa semua session pada database akan melihat data yang lama sampai data yang baru sudah di-commit ke dalam database.
5. Firebird support stored procedure dan triggers dengan bahasa yang standard sehingga tidak akan membingungkan bagi pengguna
6. Firebird bisa melakukan replikasi, solusi untuk replikasi kebanyakan dibuat oleh pihak ketiga, tetapi sebenarnya teknik replikasi ini seperti konsep trigger yang selalu memonitor adanya operasi insert, update atau delete ke dalam database.
7. Firebird support dengan multiple data file dan dapat menggunakan lebih dari satu file sebagai single logic database.
8. Software untuk mengadministrasi mudah didapat karena banyak sekali software untuk mengadministrasi database Firebird, misalnya saja EMS IB Manager, IBConsole, isql, FBManager, Marathon
9. Library connection untuk Firebird sudah tersedia seperti driver untuk ODBC, JDBC bahkan .NET database provider dan PHP
10. Memiliki banyak komunitas di internet.

**8) Microsoft SQL server**

Microsoft SQL Server adalah perangkat lunak relational database management system (RDBMS) yang didesain untuk melakukan proses manipulasi database berukuran besar dengan berbagai fasilitas. Microsoft SQL Server merupakan produk andalan Microsoft untuk database server. Kemampuannya dalam manajemen data dan kemudahan dalam pengoperasiannya membuat RDBMS ini menjadi pilihan para database administrator.

DBMS merupakan suatu system perangkat lunak untuk memungkinkan user (pengguna) untuk membuat, memelihara, mengontrol, dan mengakses database secara praktis dan efisien. Dengan DBMS, user akan lebih mudah mengontrol dan mamanipulasi data yang ada. Sedangkan RDBMS atau Relationship Database Management System merupakan salah satu jenis DBMS yang mendukung adanya relationship atau hubungan antar table. RDBMS (Relational Database Management System) adalah perangkat lunak untuk membuat dan mengelola database, sering juga disebut sebagai database engine. Istilah RDBMS, database server-software, dan database engine mengacu ke hal yang sama; sedangkan RDBMS bukanlah database. Beberapa contoh dari RDBMS diantaranya Oracle, Ms SQL Server, MySQL, DB2, Ms Access.

**9) Visual Foxpro 6.0**

Pada tahun 1984, Fox Software memperkenalkan FoxBase untuk menyaingi dBase II Ashton-Tate. Pada saat itu FoxBase hanyalah perangkat lunak kecil yang berisi bahasa pemrograman dan mesin pengolah data. FoxPro memperkenalkan GUI (Graphical Unit Interface) pada tahun 1989. FoxPro berkembang menjadi Visul FoxPro pada tahun 1995. kemampuan pemrograman prosedural tetap dipertahankan dan dilengkapi dengan pemrograman berorietasi objek. Visual FoxPro 6.0 dilengkapi dengan kemampuan untuk berinteraksi dengan produk desktop dan client/server lain dan juga dapat membangun aplikasi yang berbasis Web. Dengan adanya Visual Studio, FoxPro menjadi anggotanya. Sasaran utama Visual Studio adalah menyediakan alat bantu pemrogrman dan database untuk mengembangka perangkat lunak yang memenuhi tuntutan zaman.

Model data yang digunakan Visual FoxPro yaitu model relasional. Model Relasional merupakan model yang paling sederhana sehingga mudah di pahami oleh pengguna, serta merupakan paling popular saat ini. Model ini menggunakan sekumpulan table berdimensi dua (yang disebut relasi atau table), dengan masing-masing relasi tersusun atas tupel atau baris dan atribut. Relasi dirancang sedemikian rupa sehingga dapat menghilangkan kemubajiran data dan mengunakan kunci tamu untuk berhubungan dengan relasi lain.

**10) Database Desktop Paradox**

Database desktop merupakan suatu program "Add-Ins", yaitu program terpisah yang langsung terdapat pada Borland Delphi. Pada database desktop terdapat beberapa DBMS yang terintegrasi di dalamnya antara lain Paradox 7, Paradox 4, Visual dBase, Foxpro, Ms. SQL, Oracle, Ms. Acces, db2 dan interbase. Dari beberapa DBMS tersebut kita akan memilih salah satu yaitu Paradox yang akan dibahas lebih lanjut, khususnya Paradox 7. Dalam Paradox 7 ini, pada 1 file database hanya mengizinkan 1 tabel, berbeda dengan DBMS lain yang mengizinkan beberapa tabel pada 1 file database seperti pada Ms. Acces.

**c. Rangkuman**

Sistem manajemen basis data (*database management system*, DBMS), atau sering disingkat SMBD, adalah suatu sistem atau perangkat lunak yang dirancang untuk mengelola suatu basis data dan menjalankan operasi terhadap data yang diminta banyak pengguna.

Sebuah sistem manajemen basis data relasional atau dikenal sebagai *relational database management system (RDBMS)* adalah sebuah program komputer (atau secara lebih tipikal adalah seperangkat program komputer) yang dirancang untuk mengatur atau mengelola sebuah basis data sebagai sekumpulan data yang disimpan secara terstruktur, dan melakukan operasi-operasi data atas permintaan penggunanya.

Hukum cood adalah suatu ketentuan atau aturan dan definisi standar dari sebuah sistem basis data relasional, yang diperkenalkan oleh Edgar F. Codd. Hukum Codd terdiri dari dua belas kriteria atau ketentuan. Software atau perangkat lunak DBMS yang sering digunakan dalam aplikasi program dan sangat populer adalah MySQL, MS SQL Server, Oracle, IBM DB/2, dan

PostgreSQL DBMS yang bersifat open source: antara lain : Cloudscape, Derby, Firebird, H2, HSQLDB, Ingres, MaxDB, MonetDB, MySQL, PostgreSQL, SQLite, tdbengine

**d. Tugas : Mengamati Berbagai Ragai Jenis DBMS**

Dalam kegiatan ini peserta didik akan melakukan pengamatan melalui *teks book* secara berkelompok, satu kelompok terdiri dari dua sampai tiga orang. Dalam kegiatan ini peserta didik akan melakukan pengamatan melalui teks book terhadap berbagai ragam jenis DBMS. Bacalah seluruh langkah pengamatan dibawah ini kemudian lakukan dengan cermat dan teliti dengan perangkat yang telah disediakan.

1. Bentuk kelompok diskusi setiap kelompok terdiri dari tiga orang.
2. Dengan menggunakan fasilitas internet carilah sumber bacaan tentang berbagai ragam jenis DBMS yaitu: micorosft SQL Server, Micorsoft access, Oracle dan MySQL. Catat hasilnya (sumber bacaan) dalam bentuk tabel
3. Diskusikan dalam kelompok untuk berbagai ragam jenis DBMS sebagaimana telah disebutkan pada langkah 2. Untuk setiap DBMS diskusikan tentang: 1) Industri pembuat 2) fitur-fitur yang ada dalam DBMS 3) Diagram atau gambar arsitektur DBMS. 4) versi atau software database yang ada 5) kelebihan atau kekurangan setiap DBMS. Catat hasilnya dalam bentuk tabel.
4. Kumpulkan data-data setiap langkah dan analisis data tersebut menggunakan analisis diskriptif.
5. Komunikasikan hasilnya dalam kelompok dan buatlah kesimpulan.
6. Presentasikan hasil diskusi bersama-sama dengan kelompok lainnya dan guru pembimbing.

**e. Test Formatif.**

Dalam test ini setiap peserta didik membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

1. Jelaskan secara singkat definisi DBMS, RDBMS dan apa perbedaan keduanya ?

2. Sebutkan berbagai ragam jenis perangkat lunak DBMS baik yang bersifat komersial dan oepn source ?

**f. Lembar Jawaban Test Formatif (LJ).**

**LJ- 01 : Definis, DBMS, RDBMS dan perbedaan keduanya.**

..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................

**LJ- 02 : Berbagai ragam jenis DBMS komersial dan open source**

**g. Lembar Kerja Peserta Didik.**

## Kegiatan belajar 18: Arsitektur Aplikasi Basis data

**a. Tujuan Pembelajaran.**

Setelah mengikuti kegiatan belajar 18 ini diharapkan peserta didik dapat:

- ✓ Memahami konsep arsitektur sistem manajemen basis data
- ✓ Membedakan berbagai ragam jenis arsitektur aplikasi basis data

**b. Uraian materi**.

**1) Definisi Arsitektur aplikasi basis data**

Pengertian Arsitektur pada umumnya terkait dengan rancangan suatu bangunan atau gedung. Sebelum membangun sebuah rumah seorang akhli gambar harus membuat gambar arsitekturnya. Dari gambar rancangan rumah tersebut seorang developer dapat membangun rumah tersebut. Konsep tersebut dapat pula diterapkan untuk membangun aplikasi basis data.

Arsitektur aplikasi basis data menjelaskan rancangan dasar aplikasi basis data yang akan dibangun. Arsitektur basis data menggambarkan diagram interaksi antara komponen-komponen penyusun sistem manajemen basis data. Komponen-komponen tersebut meliputi perangkat hardware, software, jaringan komputer,pengguna dan lain-lain Berdasarkan arsitekturnya aplikasi sistem manajemen basis data (SMBD) dibedakan menjadi beberapa macam antara lain adalah sebagai berikut :

1. SMBD terpusat (CDBMS). Pada sistem ini semua proses utama dan fungsi sistem manajemen basis data sepert user application programs dan user interface programs berada secara terpusat di satu komputer berkecepatan dan kapasitas tinggi (main frame). pengguna mengakses basis data menggunakan terminal komputer.
2. SMBD terdistribusi (DDBMS) Pada sistem ini data disimpan pada beberapa tempat (*site*), setiap tempat diatur dengan suatu DBMS yang dapat berjalan secara independent. Perangkat lunak dalam sistem ini akan mengatur pendistribusian data secara transparan.
3. SMBD paralel. Dalam Sistem manajemen basis data ini menggunakan beberapa prosesor dan disk yang dirancang untuk dijalankan secara paralel dan simultan. sistem ini digunakan untuk memperbaiki kinerja dari DBMS

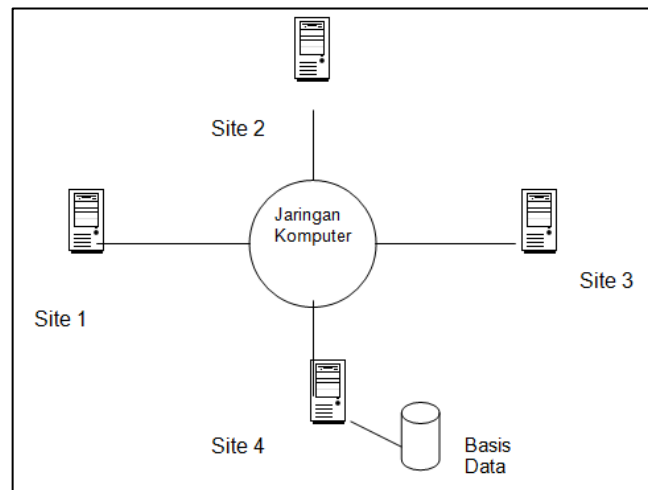
Dari tiga ragam jenis SMBD diatas terdapat beberapa model arsitektur aplikasi SMBD. Perkembangan Arsitektur SMBD cukup pesat dan cepat dengan mengikuti trend yang sejalan dengan kemajuan arsitektur sistem komputer dan teknologi informasi dan komunikasi. Beberapa ragam jenis arsitektur aplikasi SMBD tersebut antar lain ialah :

1. Arsitektur Teleprocessing
2. Arsitektur File-Server Architecture
3. Arsitektur Singgle tier
4. Arsitektur two-tier client/server
5. Arsitektur three-tier client/server.
6. Arsitektur N-tier client/server.
7. Paralel arsitektur

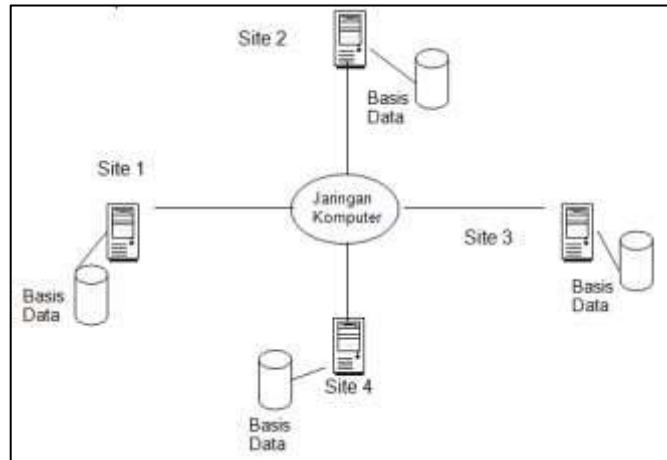
**2) Centralized Database manajemen Sistem (CDBMS)**

Pada sistem ini semua proses utama dan fungsi sistem manajemen basis data sepert user application programs dan user interface programs berada secara terpusat di satu komputer berkecepatan dan kapasitas tinggi (main frame). pengguna mengakses basis data menggunakan terminal komputer.

Arsitektur DBMS telah mengikuti trend sejalan dengan kemajuan arsitektur sistem komputer. Permulaan arsitektur DBMS dimulai dgn bentuk Arsitektur DBMS Terpusat (Centralized DBMS Architecture). Pada arsitektur ini digunakan komputer main frame yg menyediakan semua proses utama seperti fungsinya pada DBMS (user application programs & user interface programs).

Bentuk arsitektur terpusat ini menggambarkan pengaksesan terminal-terminal komputer (client) pada komputer server, berupa display informasi dan kontrol saja, karena pada terminal komputer tidak memungkinkan memiliki resource yang lebih. Seiring perkembangan teknologi dan turunnya harga hardware, banyak terminal user digantikan dengan PC, akan tetapi DBMS masih ditempatkan terpusat (Application program execution & user interface processing ditempatkan pada satu mesin). Gambar dibawah ini menjelaskan Arsitektur Centralized Database manajemen Sistem (CDBMS)

### 3) Distributed Database manajemen Sistem (DDBMS)

DDBMS memiliki satu logikal basis data yang dibagi ke dalam beberapa fragment. Dimana setiap fragment disimpan pada satu atau lebih komputer dibawah kontrol dari DBMS yang terpisah dengan mengkoneksi komputer menggunakan jaringan komunikasi. DDBMS memungkinkan direplikasi dan alokasi penyimpanan disembunyikan sehingga tidak diketahui pengguna. Pada sistem ini data disimpan pada beberapa tempat (*site*), setiap tempat diatur dengan suatu DBMS yang dapat berjalan secara independent. Perangkat lunak dalam sistem ini akan mengatur pendistribusian data secara transparan. Setiap site memiliki kemampuan untuk mengakses permintaan pengguna pada data lokal dan juga mampu untuk memproses data yang disimpan pada komputer lain yang terhubung dengan jaringan. Pengguna mengakses basis data terdistribusi dengan menggunakan dua aplikasi yaitu aplikasi lokal dan aplikasi global, sehingga DDBMS memiliki karakteristik yaitu :

- Kumpulan dari data logik yang digunakan bersama-sama
- Data di bagi menjadi beberapa fragment
- Fragment mungkin mempunyai copy ( replika )
- Fragment / replika nya di alokasikan pada yang digunakan
- Setiap site berhubungan dengan jaringan komunikasi
- Data pada masing-masing site dibawah pengawasan DBMS
- DBMS pada setiap site dapat mengatasi aplikasi lokal, secara otonomi
- Masing-masing DBMS berpastisipasi paling tidak satu global aplikasi.

Tigal hal penting yang harus terdapat pada basis data terdistribusi adalah :

- Independensi data terdistribusi : pemakai tidak perlu mengetahui dimana data berada (merupakan pengembangan prinsip independensi data fisik dan logika).
- Transaksi terdistribusi yang atomic : pemakai dapat menulis transaksi yang mengakses dan mengubah data pada beberapa tempat seperti mengakses transaksi
- Transparansi basis data terdistribusi agar terlihat sistem ini seperti basis data tersentralisasi. Hal Ini mengacu pada prinsip dasar dari DBMS (Date,1987b). Transparansi memberikan fungsional yang baik untuk pengguna tetapi mengakibatkan banyak permasalahan yang timbul dan harus diatasi oleh DDBMS.

Terdapat dua tipe basis data terdistribusi yaitu

- Homogen : yaitu sistem dimana setiap tempat menjalankan tipe DBMS yang sama
- Heterogen : yaitu sistem dimana setiap tempat yang berbeda menjalankan DBMS yang berbeda, baik Relational DBMS (RDBMS) atau non relational DBMS.

Beberapa keuntungan penggunaan DDBMS

1. Merefleksikan bentuk dari struktur organisasi. Suatu organisasi memiliki sub organisasi di lokasi yang tersebar di beberapa tempat, sehingga basis data yang digunakan tersebar sesuai lokasi dari sub organisasi tersebut.
2. Penggunaan bersama dan lokal otonomi. Distribusi secara geografis dari sebuah organisasi dapat terlihat dari data terdistribusinya, pengguna pada setiap site dapat mengakses data yang disimpan pada site lain. Data dapat dialokasikan dekat pengguna pada sebuah site, sehingga mempunyai kontrol terhadap data dan secara konsekuen dapat memperbaharui dan memiliki kebijakkan untuk data tersebut. DBA global mempunyai tanggung jawab untuk semua sistem. Umumnya sebagian dari tanggung jawab tersebut di serahkan kepada tingkat lokal, sehingga DBA lokal dapat mengatur lokal DBMS secara otonomi.
3. Keberadaan data yang ditingkatkan. Pada DBMS yang tersentralisasi kegagalan pada suatu site akan mematikan seluruh operasional DBMS. Namun pada DDBMS kegagalan pada salah satu site, atau kegagalan pada hubungan komunikasi dapat membuat beberapa site tidak dapat di akses, tetapi tidak membuat operasional DBMS tidak dapat dijalankan.
4. Keandalan ditingkatkan. Sebuah basis data dapat direplikasi ke dalam beberapa fragmen sehingga keberadaanya dapat di simpan di beberapa lokasi. Jika terjadi kegagalan dalam pengaksesan data pada suatu site karena jaringan komunikasi terputus maka site yang ingin mengakses data tersebut dapat mengakses site yang tidak mengalami kerusakan.
5. Kinerja yang ditingkatkan. Sebuah data ditempatkan pada suatu site dimana data tersebut banyak diakses pengguna. Hal ini mempunyai dampak yang baik untuk paralel DBMS yaitu memiliki kecepatan dalam pengkasesan data yang lebih baik dibandingkan dengan basis data tersentralisas.i Setiap site hanya menangani sebagian dari seluruh basis data, mengakibakan perbedaan pada pelayanan CPU dan I/O seperti yang di karakteristikan pada DBMS tersentralisasi.
6. Ekonomi. Grosch's Law menyatakan daya listrik dari sebuah komputer di hitung menurut biaya yang dihabiskan dari penggunaan peralatannya, 3 kali biaya peralatan, 9 kali dari daya listrik . Sehingga lebih murah jika

membuat sebuah sistem yang terdiri dari beberapa mini komputer yang mempunyai daya yang sama jika dibandingkan dengan memiliki satu buah super komputer. Oleh karena itu lebih efektif untuk menambah beberapa workstation untuk sebuah jaringan dibandingkan dengan memperbaharui sistem mainframe. Potensi yang juga menekan biaya yaitu menginstall aplikasi dan menyimpan basis data yang diperlukan secara geografi sehingga mempermudah operasional pada setiap situs.

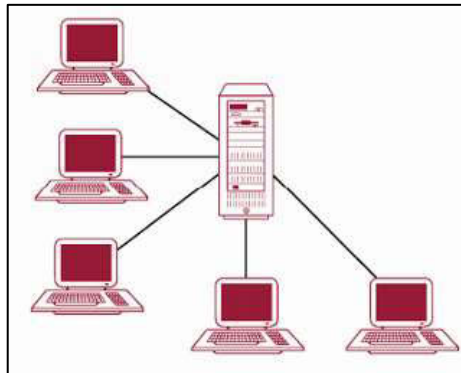
7. Perkembangan modular. Dalam sistem terdistribusi lebih mudah untuk menangani ekspansi. Site baru dapat di tambahkan ke suatu jaringan tanpa mempengaruhi operational site yang ada. Penambahan ukuran basis data dapat ditangani dengan menambahkan pemrosesan dan daya tampung penyimpanan pada suatu jaringan.

Kelemahan atau kekurangan DDBMS yaitu :

1. Kompleksitas. Pada DDBMS yang digunakan adalah replikasinya, yang asli tidak digunakan untuk operasional. Hal ini untuk menjaga reliabilitas dari suatu data. Hal ini juga menimbulkan masalah yang sangat kompleks dimana DBA harus dapat menyediakan pengaksesan dengan cepat, keandalan dan keberadaan basis data yang up to date. Jika hal itu tidak terpenuhi, akan terjadi penurunan kinerja, keandalan dan kerberadaan dari DBMS tersebut.
2. Biaya. Meningkatnya kompleksitas DDBMS berarti biaya untuk perawatan lebih besar dibandingkan dengan DBMS tersentralisasi, seperti biaya untuk membuat jaringan, biaya komunikasi yang berjalan, orang-orang ahli dalam penggunaan, pengaturan dan pengawasan dari DDBMS.
3. Keamanan. Pada DBMS tersentralisasi, pengaksesan data lebih terkontrol. Sedangkan pada DDBMS bukan hanya replikasi data yang harus di kontrol tetapi jaringan juga harus dapat di kontrol keamanannya. Pengontrolan Integritas lebih sulit Kesatuan basis data yang mengacu pada keabsahan dan kekonsistenan dari data yang disimpan. Kesatuan biasanya di ekspresikan pada batasan, dimana berisi aturan untuk basis data yang tidak boleh diubah. Membuat batasan untuk integrity, umumnya memerlukan pengaksesan ke sejumlah data yang sangat besar untuk mendefinisikan batasan tersebut, namun hal ini tidak termasuk di dalam operasional update itu sendiri. Dalam DDBMS, komunikasi dan biaya

pemrosesan yang dibutuhkan untuk membuat suatu batasan integrity mungkin tidak diperbolehkan.

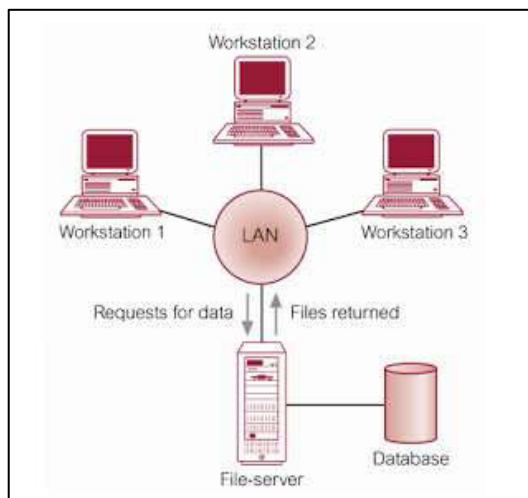
**4) Teleprocessing Arsitektur**

Teleprocessing adalah suatu arsitektur tradisional untuk multi-user system, dimana sebuah CPU terhubung dengan beberapa workstation. Pada Arsitektur ini semua pemrosesan dikerjakan dalam batasan fisik komputer yang sama.

Terminal untuk pemakai berjenis 'dumb', yang tidak dapat berfungsi sendiri dan masing-masing dihubungkan ke komputer pusat. Terminal-terminal tersebut mengirimkan pesan melalui subsistem pengontrol komunikasi pada sistem operasi ke program aplikasi, yang bergantian menggunakan layanan DBMS. Dengan cara yang sama, pesan dikembalikan ke terminal pemakai. Arsitektur ini menempatkan beban yang besar pada komputer pusat yang tidak hanya menjalankan program aplikasi tetapi juga harus menyelesaikan sejumlah pekerjaan pada terminal seperti format data untuk tampilan di monitor. Beberapa kelebihan arsitektur sistem basis data ini antara lain ialah :

1. Murahnya biaya telekomunikasi sekarang ini memungkinkan terminal terminal untuk saling berhubungan untuk membentuk real time systems.
2. Memungkinkan pengguna bersama–sama menggunakan komputer.
3. Komputer akan membagi waktunya bergantian untuk tiap – tiap pemakai.
4. Dengan time sharing systems sekarang ini sebuah komputer pusat dapat melayani sampai ribuan terminal.
5. Semua data dan program aplikasi tersimpan di hardisk komputer pusat.

Sedangkan Kekurangannya adalah : Membutuhkan Komputer Besar atau Mainframe yang harganya sangat mahal.

5) **File-Server Architecture**

File Server Architecture adalah merupakan suatu komputer server yang dihubungkan dengan beberapa workstation melalui suatu jaringan (network). Database diletakkan pada file-server. DBMS dan aplikasi dijalankan pada setiap workstation. Meskipun aplikasi dan DBMS dijalankan pada setiap workstation tetapi tetap meminta file dari file server jika diperlukan. Dengan cara ini,file server berfungsi sebagai sebuah hard disk yang digunakan secara bersamaan.

Kelebihannya arsitektur file-server ini adalah:

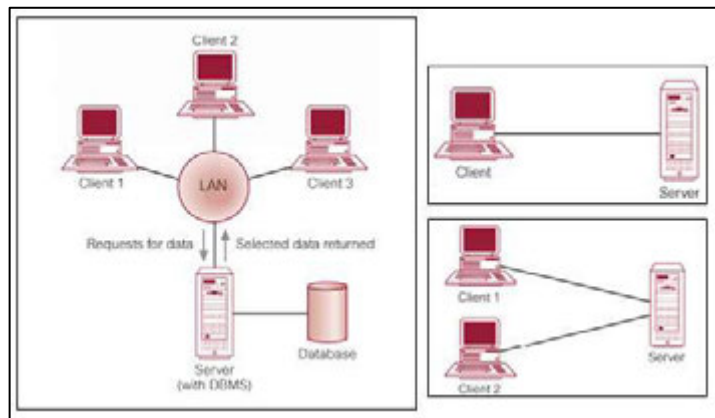
1. Dapat membagi berbagai jenis seperti dokumen, spreadsheet, gambar dan database.
2. File server menyimpan file-file yang dibutuhkan oleh aplikasi dan DBMS.
3. Aplikasi dan DBMS bekerja pada masing-masing workstation, meminta file pada file server ketika dibutuhkan.
4. File server bertindak sebagai pengelola file dan memungkinkan klien mengakses file tersebut.

Sedangkan Kekurangannya ialah:

1. Terdapat lalulintas jaringan yang besar sehingga mengakibatkan kepadatan jaringan.
2. Masing-masing workstation membutuhkan salinan DBMS.
3. Kontrol terhadap *concurrency, recovery dan integrity* menjadi lebih kompleks karena sejumlah DBMS mengakses file secara bersamaan.
4. Beban jaringan tinggi karena tabel yang diminta akan diserahkan oleh file server ke klien melalui jaringan.

6) **Client-Server Architecture**

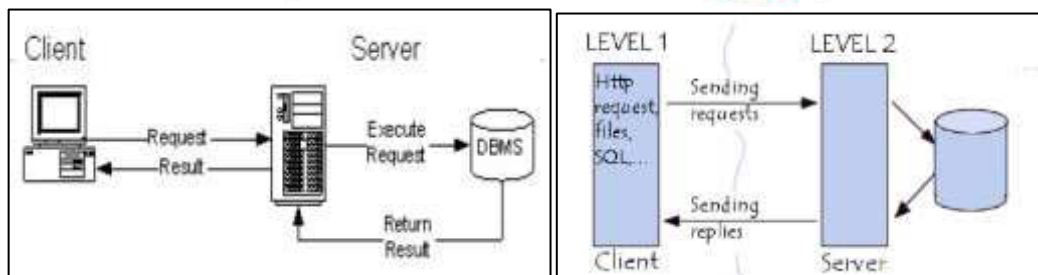
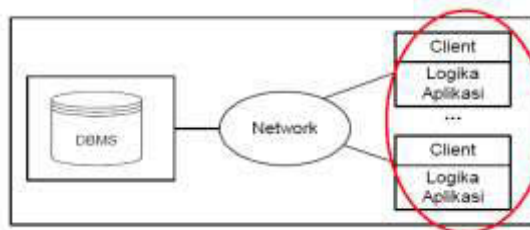
Untuk mengatasi kelemahan arsitektur file server dikembangkan arsitektur client-server. Client-server menunjukkan cara komponen software berinteraksi dalam bentuk sistem. Dalam hal ini server menangani database dan DBMS. Sementara itu Client mengatur user interface dan menjalankan aplikasi.

Konsep arsitektur client/server mengasumsikan sebuah kerangka dasar (framework) yang terdiri atas banyak PC yang terhubung melalui LAN beserta tipe-tipe jaringan komputer lainnya. Suatu **Client** adalah mesin user yang menyediakan kemampuan user interface dan local processing. Suatu **Server** adalah mesin yang menyediakan berbagai service ke mesin client (file access, printing, archiving, or database access). Ada kemungkinan suatu mesin hanya menginstall software client saja, yang lain software server, atau bahkan keduanya pada satu mesin (seperti pada gambar physical client/server sebelumnya). Dua arsitektur DBMS yang mendasari framework client/server: *two-tier client/server* dan *three-tier client/server.*

Tugas dari komputer *Client* adalah: 1) Mengatur user Interface. 2) Menerima dan memeriksa syntax input dari user. 3) Membangun (Generates) permintaan DB dan mengirimkannya ke server. 4) Memberikan respon balik ke user. Sedangkan tugas dari komputer *server* adalah : 1) Menerima & memroses permintaan DB dari client. 2) Memeriksa autorisasi. 3) Menjamin batasan integritas data. 4) Menampilkan queri/proses update dan mengirimkannya ke user. 5) Memelihara System Catalog. 6) Menyediakan kontrol recovery. 7) Menyediakan akses basis data yang akurat. Kelebihan dari sistem arsitektur client-server ini ialah :

1. Klien bertanggung jawab dalam mengelola antar muka pemakai (mencakup logika penyajian data, logika pemrosesan data, logika aturan bisnis).
2. Database server bertanggung jawab pada penyimpanan, pengaksesan, dan pemrosesan database.
3. Otentikasi pemakai, pemeriksaan integrasi, pemeliharaan data dictionary dilakukan pada database server.
4. Akses yang lebih luas terhadap database.
5. Meningkatkan performa dan konsistensi.
6. Pengurangan biaya hardware, biaya komunikasi dan beban jaringan

Sedangkan kekurangan arsitektus sistem basis data ini ialah: Database server dituntut memiliki kemampuan pemrosesan yang tinggi.

**7) *Arsitektur two tier client server***

Pada dasarnya arsitektur two-tier sering disebut sebagai arsitektur client/server, yang terdiri komputer client dan komputer server, yang berinteraksi melalui protokol yang sifatnya well-defined. Dalam arsitektur client/server tradisional, client hanya mengimplementasikan GUI (Graphical User Interface), sedang server hanya mengimplementasikan logika bisnis dan manajemen data. Client tersebut disebut thin client (klien tipis). Pada bentuk lain, terdapat juga client yang lebih powerfull dengan mengimplementasikan GUI dan logika bisnisnya sedang sisanya pada sisi server, yang disebut thick client (klien tebal).

Model thick client memiliki beberapa kelemahan:

1. Tidak memiliki tempat pusat untuk memperbaharui dan memelihara logika bisnis, karena berjalan pada sisi client.
2. Rasa saling percaya antara client-server (strore procedure pada client),
3. Tidak dapat menangani jumlah client yang besar.
4. Thick client tidak diskalakan seiring dengan pertambahan akses aplikasi dan sistem database.

Pada database client/server, saat pengaksesan DBMS dibutuhkan: program membuka koneksi ke DBMS server, sekali koneksi terbuat maka program client dapat berkomunikasi dengan DBMS. Contoh: ODBC (Open Database Connectivity) yang menyediakan API (Application Programming Interface), JDBC, yg digunakan program client Java utk akses ke DBMS.Interaksi antara client dan server selama pemrosesan query SQL adalah sebagai berikut :

1. Client melakukan parsing query pemakai dan memecahnya ke dalam sejumlah query independent untuk setiap tempat. Setiap query tersebut dikirim ke server yang sesuai.
2. Setiap server memproses query lokal dan mengirim relasi hasil ke client.
3. Client mengkombinasikan hasil sub query untuk memproduksi hasil dari query asal yang dikirim.

Pada pendekatan tersebut Server SQL: juga disebut transaction server (database processor (DP) / back-end machine / DBMS), sedangkan Client : disebut application processor (AP) atau front-end machine.

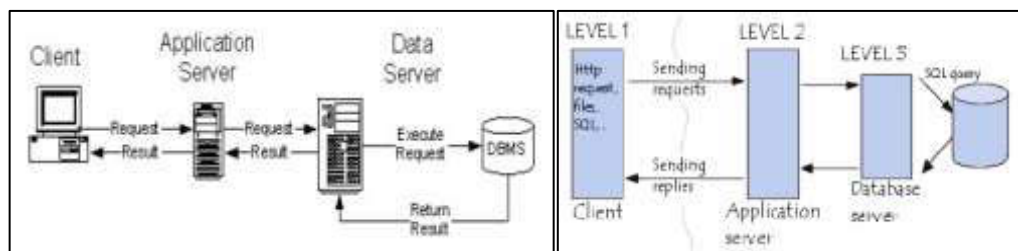
**8) Arsitektur Three-Tier Client/Server**

*Three Tier Architecture* merupakan inovasi dari arsitektur client-server. Pada arsitektur Three-tier ini terdapat application server yang berdiri di antara client dan database server. Contoh dari application server adalah IIS (Internet Information Services), WebSphere, dan sebagainya. Arsitektur ini memisahkan antara logika aplikasi dari manajemen data, yang meliputi:

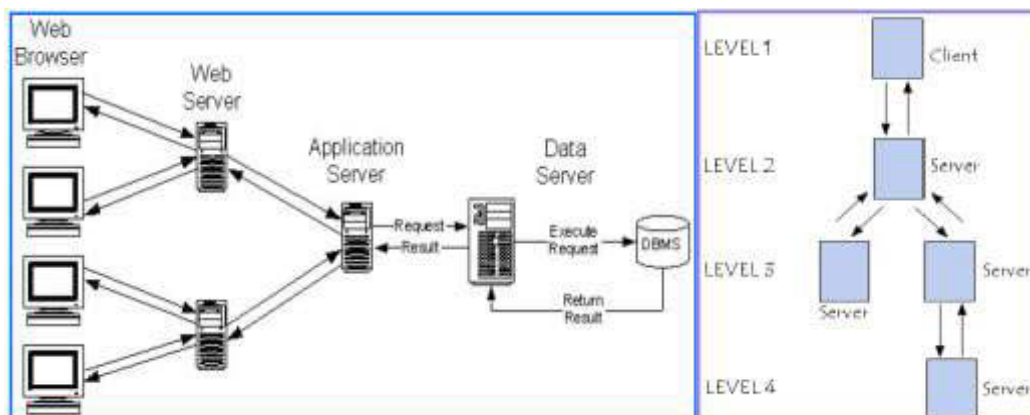
1. Presentation Tier (Client) Berisi interface natural yang dibutuhkan user untuk membuat request, menyediakan input dan melihat hasil. (GUI)
2. Middle Tier (Application Layer/Web Server) Berisi logika aplikasi untuk dieksekusi, berbagai macam kode program (C++, Java, dll) sebagai proses bisnis logic yang kompleks. (Application Programs, Web Pages).

3. Data Management Tier (Database Server) Berisi DBMS.

Beberapa keuntungan arsitektur three-tier adalah :

1. Sistem Heterogen Mendukung pada berbagai platform yang berbeda. Sehingga modify code pada suatu tier tidak berdampak pada tier yang lain.
2. Thin Client  Membutuhkan kemampuan komputasi yang memadai pada presentation layer (biasanya web browser).
3. Akses Data Terintegrasi. Kemampuan middle tier dalam mengatur koneksi pada semua sistem database yang terlibat secara terpusat.
4. Stabilitas pada Banyak Client , yaitu kemampuan middle tier dalam mengatur hubungan database pada client.
5. Kemudahan dalam Pengembangan Software  Masing-masing tier dapat dikembangkan lebih lanjut (debug, test) tanpa mempengaruhi yang lain.

**Gambar 54. Aarsitektur three-tier client server**

### 9) Arsitektur N-tier atau multi tier.

Istilah arsitektur ini muncul karena dalam implementasi aplikasi basis data base dimungkinkan suatu aristektur aplikasi terdiri dari banyak tier .

Salah satu contoh aplikasi basis datayang menggunakan arsitektur ini ialah situs amazon .com, dimana pelanggan internet dapat memesan buku secara online. Pelanggan dapat melihat katalog buku amazon.com yang sebenarnya ada pada database amazon.com. Jika pelanggan ingin memesan salah satu buku, maka pelanggan tersebut perlu memasukkan informasi mengenai dirinya dan yang terlebih penting adalah data mengenai kartu kreditnya. Untuk dapat memesan buku data kartu kredit pelanggan tersebut harus divalidasi terlebih dahulu: seperti kode PIN, masa berlaku kartu, limit kredit. Setelah dinyatakan valid maka pelanggan dapat melakukan transaksi pemesanan buku.

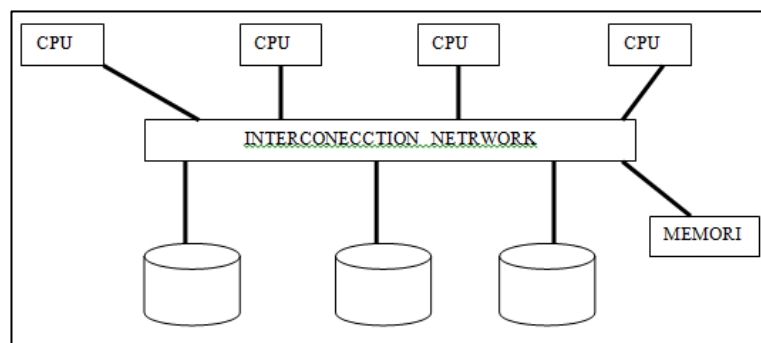
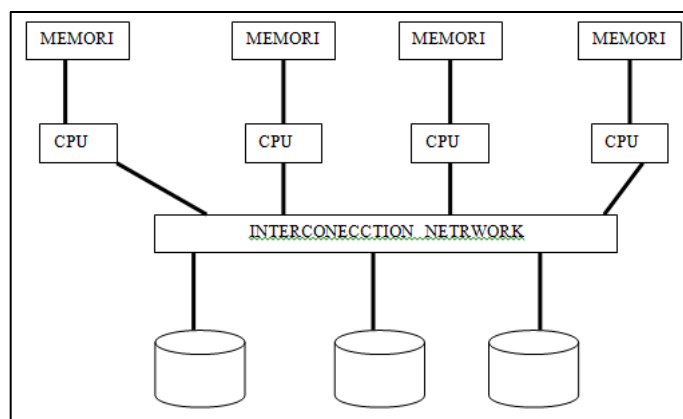
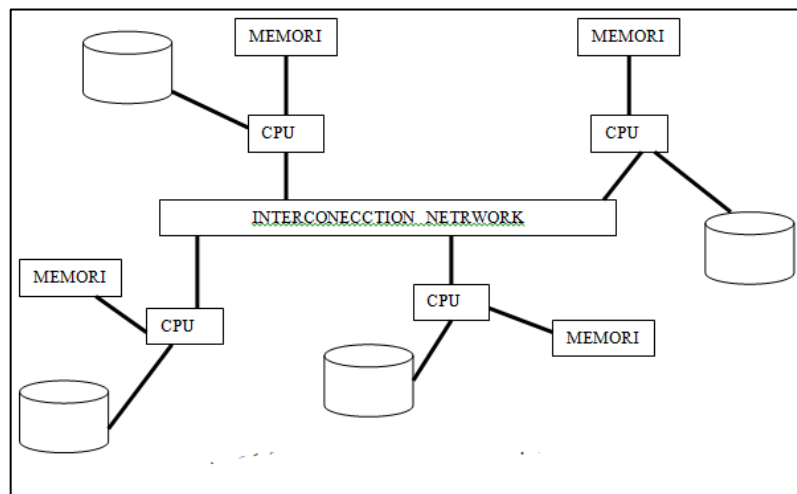
**10) Arsitektur Paralel**

Sistem manajemen basis data ini menggunakan beberapa prosesor dan disk yang dirancang untuk dijalankan secara paralel. Arsitektur ini digunakan untuk memperbaiki kinerja dari DBMS. Paralel DBMS di jalankan oleh berbagai multi prosesor. Paralel DBMS menghubungkan beberapa mesin yang berukuran kecil untuk menghasilkan keluaran sebuah mesin yang berukuran besar dengan skalabilitas yang lebih besar dan keandalan dari basis datanya.

Untuk menopang beberapa prosesor dengan akses yang sama pada satu basis data, DBMS paralel harus menyediakan manajemen sumber daya yang dapat diakses bersama. Sumber daya apa yang dapat digunakan bersama, dan bagaimana sumber daya tersebut diimplementasikan. Hal ini mempunyai efek langsung pada kinerja dan skalabilitas dari sistem dan tergantung dari aplikasi atau lingkungan yang digunakan. Terdapay tiga arsitektur yang digunakan pada paralel DBMS yaitu :

a. Penggunaan memory bersama ( *share memory* )
b. Penggunaan disk bersama ( *share disk* )
c. Penggunaan secara sendiri-sendiri (*share nothing)*

Arsitektur Penggunaan Memori Bersama ( *Share Memory* ) adalah sebuah arsitektur yang menghubungkan beberapa prosesor di dalam sistem tunggal yang menggunakan memori secara bersama – sama. Arsitektur ini dikenal

dengan SMP (Symmetric Multiprocessing ), metode ini sering digunakan dalam bentuk workstation personal yang mensupport beberapa mikroprosesor dalam paralel DBMS, RISC ( Reduced Instruction Set Computer ) yang besar berbasis mesin sampai bentuk mainframe yang besar. Arsitektur ini menghasilkan pengaksesan data yang sangat cepat yang dibatasi oleh beberapa prosesor, tetapi tidak dapat digunakan untuk 64 prosesor dimana jaringan komunikasi menjadi masalah ( terjadinya bottleneck). Gambar dibawah ini menjelaskan arsitektur paralel dengan penggunaan memory bersama

Arsitektir Penggunaan Disk Bersama ( *Share Disk* ) adalah sebuah arsitektur yang mengoptimalkan jalannya suatu aplikasi yang tersentrallisasi dan membutuhkan keberadaan data dan kinerja yang tinggi. Setiap prosesor dapat mengakses langsung semua disk, tetapi prosesor tersebut memiliki memori sendiri. Seperti halnya penggunaan secara sendiri–sendiri. Arsitektur ini menghapus masalah pada penggunaan memori bersama tanpa harus mengetahui sebuah basis data di partisi. Arsitektur ini di kenal dengan *cluster.* Gambar dibawah ini menjelaskan arsitektur paralel dengan penggunaan disk bersama.

Arsitektur Penggunaan Secara sendiri – sendiri (*Share nothing*) hampir sama dengan DBMS terdistribusi, namun pendistribusian data pada paralel DBMS hanya berbasis pada kinerjanya saja. Sering di kenal dengan *Massively parallel processing* ( MPP ) yaitu arsitektur dari beberapa prosesor di mana setiap prosesor adalah bagian dari sistem yang lengkap yang memiliki memori dan disk. Basis data ini di partisi untuk semua disk pada setiap sistem yang berhubungan dengan basis data. Ddata diberikan secara transparan untuk semua pengguna yang menggunakan sistem . Arsitektur ini lebih dapat di hitung skalabilitasnya dibandingkan dengan share memory dan dengan mudah dapat mensupport prosesor yang berukuran besar. Kinerja dapat optimal jika data di simpan di lokal DBMS. Gambar dibawah ini menjelaskan arsitektur paralel dengan penggunaan secara mandiri.

**c. Rangkuman**

Arsitektur aplikasi basis data menjelaskan rancangan dasar aplikasi basis data yang akan dibangun. Arsitektur basis data menggambarkan diagram interaksi antara komponen-komponen penyusun sistem manajemen basis data. Berdasarkan arsitekturnya aplikasi sistem manajemen basis data (SMBD) dibedakan menjadi: 1) Sistem manajemen basis data terpusat terpusat (CDBMS). 2) Sistem manajemen basis data terdistribusi (DDBMS). 3) paralel DBMS. Dari tiga ragam jenis SMBD diatas terdapat beberapa model arsitektur yaitu :1) Arsitektur Teleprocessing. 2) Arsitektur File-Server Architecture. 3)

Arsitektur Singgle tier. 4) Arsitektur two-tier client/server. 5) Arsitektur three-tier client/server. 6) Arsitektur N-tier client/server. 7) Paralel arsitektur

Pada CDBMS semua proses utama dan fungsi sistem manajemen basis data sepert user application programs dan user interface programs berada secara terpusat di satu komputer berkecepatan dan kapasitas tinggi (main frame). pengguna mengakses basis data menggunakan terminal komputer. DDBMS memiliki satu logikal basis data yang dibagi ke dalam beberapa fragment. Dimana setiap fragment disimpan pada satu atau lebih komputer dibawah kontrol dari DBMS yang terpisah dengan mengkoneksi komputer menggunakan jaringan komunikasi. Paralel DBMS menggunakan beberapa prosesor dan disk yang dirancang untuk dijalankan secara paralel. Arsitektur ini digunakan untuk memperbaiki kinerja dari DBMS. Paralel DBMS di jalankan oleh berbagai multi prosesor

### d. Tugas : Mengamati Berbagai Ragai Jenis Arsitektur Basis Data

Dalam kegiatan ini peserta didik akan melakukan pengamatan secara berkelompok, satu kelompok terdiri dari dua sampai tiga orang. Peserta didik akan melakukan pengamatan terhadap teks book terhadap berbagai ragam jenis arsitektur aplikasi DBMS. Bacalah seluruh langkah pengamatan dibawah ini kemudian lakukan dengan cermat dan teliti dengan perangkat yang telah disediakan.

1. Bentuk kelompok diskusi setiap kelompok terdiri dari tiga orang.
2. Dengan menggunakan fasilitas internet carilah sumber bacaan tentang berbagai ragam jenis arsitektur aplikasi DBMS. Catat hasilnya (sumber bacaan) dalam bentuk tabel
3. Diskusikan dalam kelompok untuk berbagai ragam jenis arsitektur aplikasi DBMS Untuk setiap arsitektur diskusikan tentang: 1) diskripsi singkat 2) Gambar atau diagram arsitektur 3) kelebihan atau keuntungan setiap arsitektur 4) kekurangan atau kelemahan setiap arsitektur. 5) Contoh aplikasi databasenya..
4. Kumpulkan data-data setiap langkah dan analisis data tersebut menggunakan analisis diskriptif.
5. Komunikasikan hasilnya dalam kelompok dan buatlah kesimpulan.

6. Presentasikan hasil diskusi bersama-sama dengan kelompok lainnya dan guru pembimbing.

**e. Test Formatif.**

Dalam test ini setiap peserta didik membaca dengan cermat dan teliti setiap butir soal dibawah ini. Kemudian berdasarkan uraian materi diatas tulislah jawabannya pada lembar jawaban test formatif yang telah disediakan.

1. Jelaskan secara singkat perbedaan antara DDBMS, CDBMS dan *Paralel database* ?
2. Jelaskan secara singkat istilah-istilah berikut: back-end, front end, ODBC, dan level dalam arsitektur aplikasi DBMS ?

**f. Lembar Jawaban Test Formatif (LJ).**

**LJ- 01 : Perbedaan antara: CDBMS, DDBMS dan paralel DBMS.**

........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................
........................................................................................................

g. **Lembar Kerja Peserta Didik.**

## Daftar Pustaka

Ramakrishnan , Ragu dan Gehrke Johannes, (2004), "Sistem manajemen Basis data" Edisi 3, terjemahan, Mc Graw Hill Education, diterbitkan ulang ulang Penerbit Andi,

Kusrini, (2007) "Strategi perancangan dan pengelolaan basis data", penerbit Andi, Yogyakarta

Ramon A, Mata Toledo dan Pauline K, Cushman, (2007), " *Schaum Outlines Dasar Dasar Data Base Relasional ",* terjemahan MC Graw Hill Education, Diterbitkan ulang oleh Penerbit Erlangga, Jakarta.